

98 P 2825



① BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

⑫ **Offenlegungsschrift**
⑩ **DE 196 54 766 A 1**

⑤ Int. Cl. 6:
G 06 F 13/12
G 06 F 3/14

⑳ Aktenzeichen: 196 54 766.0
㉑ Anmeldetag: 30. 12. 96
㉒ Offenlegungstag: 3. 7. 97

B 7

DE 196 54 766 A 1

③ Unionspriorität:
29.12.95 US 08/581168

⑦ Anmelder:
Wyse Technology, Inc., San Jose, Calif., US

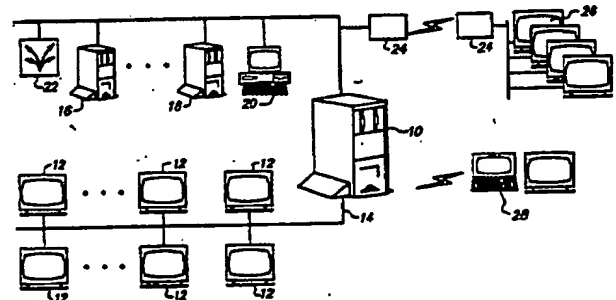
⑦A Vertreter:
STRASSE & HOFSTETTER, 81541 München

⑦Z Erfinder:

Buswell, Randy, Milpitas, Calif., US; Fox, Carol A.,
Fremont, Calif., US; Gay, Bill, Piedmont, Calif., US;
Lam, Sui M., Fremont, Calif., US; Schwebke, Curtis,
San Jose, Calif., US; Wey, Yih-Shyan, Fremont,
Calif., US

⑤A Verfahren und Vorrichtung zum Darstellen von Fenster-Anwendungsprogrammen auf einem Terminal

⑤B Es wird ein Video-Anzeige-Terminal beschrieben, das den Betrieb mit einer graphischen Benutzerschnittstelle, beispielsweise WINDOWS®, ermöglicht, und Fensterfunktionen bereitstellt, um die Verwendung von auf einem Server residenten gewöhnlichen Anwendungsprogrammen bereitzustellen, wobei hierzu nur die Anwendungsdaten vom Server zu übertragen sind sowie Tastatur- und Maus-Informationen vom Terminal an den Server. Weiterhin wird ein Verfahren zum Aktualisieren von Terminal-Betriebs-Charakteristiken über eine Kommunikationsverbindung zu einem Host beschrieben. Schließlich wird ein Verfahren und eine Vorrichtung zur Implementierung einer Vielzahl von Merkmalsgruppen in einem Terminal sowie das Hin- und Herschalten zwischen derartigen Merkmalsgruppen beschrieben.



DE 196 54 766 A 1

Beschreibung

Die vorliegende Erfindung bezieht sich allgemein auf ein Verfahren und eine Vorrichtung zum Darstellen von Informationen auf einem Terminal oder an einer Arbeitsstation und insbesondere auf ein Verfahren und eine Vorrichtung zum Formatieren und Darstellen von Informationen auf einem Terminal, auf eine graphische Benutzerschnittstelle, wie beispielsweise der Microsoft Windows®-Benutzerumgebung, einschließlich der Anwenderprogramme in dieser Umgebung.

Graphische Benutzerschnittstellen, wie die Microsoft Windows®-Bedienungsumgebung, stellen die meistverbreitete und weltweit am meisten verkaufte Anwendungs-Software dar. Solche Umgebungen werden deshalb besonders bevorzugt, weil sie leicht zu bedienen sind, einheitliche Benutzerschnittstellen bieten, eine hohe Darstellungsqualität haben und aus vielen weiteren Gründen.

Jedoch bedingen diese Benutzerumgebungen ausgebaute Arbeitsplätze und Microcomputer, wie beispielsweise Personalcomputer (PC). Diese Arbeitsplätze und Microcomputer sind zwar flexibel, verursachen aber Schwierigkeiten hinsichtlich der Sicherheit, der Zuverlässigkeit, des Bedienungskomforts und sind aufwendig. Demgegenüber sind Datenterminals im Vergleich zu Microcomputern für ihre Vorteile der besseren Sicherheit bekannt und für ihren Bedienungskomfort sowie dafür, daß sie üblicherweise preiswerter sind; diese Terminals sind jedoch üblicherweise nicht mit den am meisten benutzten graphischen Benutzerschnittstellen kompatibel. Diejenigen Terminals, die in der X-Umgebung arbeiten, bieten im beschränkten Umfang graphische Benutzerschnittstellen-Möglichkeiten unter UNIX®, sind aber teuer und erfordern einen großen Speicherplatz und sind zu den verbreiteten Windows®-Umgebungen nicht kompatibel.

Eine andere bekannte Option bietet das sogenannte plattenlose PC-System ("diskless PCS"). Das plattenlose PCS leidet jedoch unter einigen Unzulänglichkeiten. In den meisten Fällen zeigt das plattenlose PCS, das in einer "Client-Server"-Umgebung arbeitet, eine Anwendungsprogramm-Information an, indem die Anwendung vom Server heruntergeladen wird, und diese Anwendung dann lokal ausgeführt wird. Dies erfordert vom plattenlosen PC zumindest eine, wie auch immer geartete, Ausstattung, die für jede dieser Ausführungen dies auch ermöglicht. In der heute üblichen Umgebung kann dies einen Speicher von mehr als acht Megabyte, einen starken Prozessor usw. erfordern, wodurch der einzelne plattenlose PC wieder sehr teuer wird. Schließlich ist das plattenlose PCS wenig sicher und kann eine umfängliche Verwaltung erforderlich machen.

Das Windows® NT-Betriebssystem stellt eine robuste Netzwerk-Client/Server-Umgebung dar und bietet gleichzeitig Kompatibilität mit der Anwendungsprogrammebene der verbreiteten Windows® Umgebung. Jedoch wurde das NT-Betriebssystem für PC-Clients und nicht für Terminals ausgelegt. Dies führt dazu, daß die NT-Clients wiederum robust und deshalb teuer sein müssen. Außerdem ist Windows® NT für die "Client-Server"-Umgebung und nicht für eine "Multiuser"-Umgebung ausgelegt. Das kürzlich von Citrix Systems, Inc. vorgestellte WinFrame®-Betriebssystem modifiziert das Windows® NT-Betriebssystem durch Ausweitung in eine Multiuser-Umgebung, während frühere Varianten von WinFrame® anstelle von Terminals PCS-Clients benötigten.

Es besteht daher nach wie vor die Aufgabe, ein preiswertes Terminal verfügbar zu machen, das zuverlässig arbeitet, leicht zu bedienen sowie sicher ist und die Anwender-Programm-Informationen in einer Multiuser Windows®-Betriebsumgebung anzeigen kann.

Die vorliegende Erfindung bietet eine elegante Lösung, um die Nachteile des Standes der Technik zu überwinden, indem ein preiswertes Terminal für die Darstellung der mit einer Fensterumgebung kompatiblen Anwender-Software verfügbar wird.

Insbesondere bietet die Erfindung ein Display-Terminal, das in der Lage ist, in einem Multiuser-System mit einem Anwendungs-Server zu kommunizieren. Dies schafft einen sicheren Zugriff zu den Windows®-Anwendungen am Arbeitsplatz. In einem Konfigurationsbeispiel kann ein Anwendungs-Server in Form eines beliebigen Computers bereitgestellt werden, auf dem das WinFrame®-Betriebssystem läuft. Das WinFrame®-Betriebssystem der Citrix Systems, Inc. schließt das Windows® NT-Betriebssystem und zusätzliche Erweiterungen ein, die ein als ICA-3 bekanntes Anzeigeprotokoll sowie Multiuser-Fähigkeiten implementieren.

Das Terminal baut in einem Ausführungsbeispiel auf eine Hardware-Architektur mit der INTEL® X86 Prozessor-Linie auf. Zusätzlich hat das Terminal nur einen begrenzten Hauptspeicher und ist grundsätzlich nicht dafür ausgelegt, lokal moderne Anwendungsprogramme wie Textverarbeitungsprogramme, Graphikprogramme oder Datenbankprogramme oder andere populäre Programme laufen zu lassen, nicht einmal das Windows®- oder DOS-Betriebssystem selbst. Auf diese Weise unterscheidet sich ein Terminal gemäß der vorliegenden Erfindung grundsätzlich von dem bekannten X-Terminal oder dem plattenlosen PCS oder anderen PC-Systemen, die in einer Client-Server-Umgebung arbeiten.

Es ist wichtig, daß die neue Hardware-Architektur nicht auf den üblichen IBM PC/AT-Bus zurückgreift und die Firmware (die in einem ROM gespeicherte Software) innerhalb des Terminals greift weder auf den PC/AT BIOS-Standard noch auf PC-kompatible Platten-Betriebssysteme zurück. Die Terminal-Firmware implementiert Netzwerk-Zugriffserweiterungen, die mit dem Anwendungs-Server kompatibel sind, wiederum z. B. die von Citrix Systems verfügbaren ICA-3-Erweiterungen. Es wird sowohl ein hochauflösende Bildschirm zur Erleichterung der Benutzung bereitgestellt, der ein Schwarz-Weiß-Bild mit vielen Graustufen oder Farbe bietet, als auch die für eine Windows®-Umgebung typischen Eingabe- und Ausgabeeinrichtungen, wie Maus, Tastatur, Berührungsschirm und andere Ein- und Ausgabe-Einrichtungen.

Zusätzlich ist eine Netzwerkschnittstelle vorhanden, die mit dem Anwendungs-Server über konventionelle R5232-Verbindungen, Ethernet-Anschlüsse, drahtlose Verbindungen, ISDN, Glasfaseroptik, Wechselstromanschlußmodems, Kabel oder sonstwie kommuniziert. Sobald die Verbindung zum Server hergestellt ist, zeigt das Terminal die Windows® NT oder Windows 95®-Umgebung an, einschließlich beliebiger Anwendungsprogramme, die auf dem Server ausgeführt werden können und auf die der Benutzer am Terminal zugreifen kann. In einer

beispielhaften Anordnung stellt das Terminal dem Benutzer im wesentlichen dasselbe dar wie ein wesentlich teurer, weniger sicherer und schwieriger zu handhabender Personalcomputer. Im Ergebnis bietet die vorliegende Erfindung zahlreiche Merkmale, die normalerweise mit einem Multiuser-System assoziiert werden, während gleichzeitig auch viele wünschenswerte Merkmale dargeboten werden, die für eine Client-Server-Umgebung typisch sind.

Ein Merkmal der vorliegenden Erfindung ist die Möglichkeit innerhalb der Windows®-Umgebung clientenunabhängig zu arbeiten. Auf diese Weise definiert ein Benutzer mit einem System gemäß der vorliegenden Erfindung seine Betriebsumgebung und diese Umgebung folgt ihm oder ihr durch das ganze System. Ein Benutzer kann sich deshalb in ein Terminal einloggen, eine Umgebung definieren und dann sich in ein weiteres Terminal einloggen. Das zweite Terminal wird dann automatisch die Umgebung anzeigen, die der Benutzer auf dem ersten Terminal definiert hat.

Zusätzlich zu der Fähigkeit, die Anwendungsprogramm-Information anzuzeigen, verfügt das Terminal gemäß der vorliegenden Erfindung über einen Einstellmodus, welcher es dem Benutzer ermöglicht, eine Vielzahl von Betriebsarten des Terminals zu konfigurieren.

Um Terminals gemäß der vorliegenden Erfindung in die Lage zu versetzen, den von denjenigen Servern, die unter WinFrame® laufen, ausgesandten Befehlen zu folgen, ist ein besonderes Terminal-Betriebssystem entwickelt worden, das ein konventionelles PC-Betriebssystem entweder emuliert oder auf andere Weise die Anrufe beantwortet. Das Terminal-Betriebssystem schließt einen Boot-Block ein, um das System zu initialisieren und zu starten, worauf ein Kernel folgt, welcher zusätzliche Treiber und Befehls-Software lädt, und zwar einschließlich eines modifizierten WinFrame®-Clientencodes. Die Ausführung des modifizierten WinFrame®-Clients schließt die Herstellung einer Verbindung zu einem Anwendungs-Server ein.

Zusätzlich zu der Fähigkeit, eine Anwendungsinformation in einer Microsoft Windows®-Umgebung anzuzeigen, bietet die vorliegende Erfindung auch die Fähigkeit, mit anderen Servern oder Host-Rechnern, auf denen von Windows® abweichende Betriebssysteme laufen, dadurch zu kommunizieren, daß andere residente Terminal-Emulationen oder Kenndaten eingeschlossen sind, mit der Möglichkeit, durch einfachen Knopfdruck zwischen den verschiedenen Kenndaten hin- und herzuschalten.

Ein weiteres Merkmal der vorliegenden Erfindung besteht in der Implementierung einer graphischen Benutzerschnittstelle für die Konfiguration eines Terminals und anderer lokal zu lösender Tasks. Insbesondere stellt die graphische Schnittstelle eine Vielzahl von Fenstern zur Verfügung und jedes Fenster ermöglicht eine erneute Konfiguration von einem oder mehreren Terminal-Kennzeichen oder anderer lokaler Tasks. In jedem Fenster sind die Auswahlmöglichkeiten in Gruppen angeordnet, die in diesem Fenster konfiguriert werden können. Dieses Auswählen oder diese Wahlmöglichkeiten, von denen der Benutzer Gebrauch machen kann, werden angezeigt oder sind durch ein sogenanntes "Pull down"- oder ein ähnliches Menü anzeigbar. Verschiedene Datenstrukturen gehören zu verschiedenen Arten von Gruppen oder Wahlmöglichkeiten.

Ein anderes Merkmal der vorliegenden Erfindung ist der Einschluß des Terminal-Betriebssystems in einen Kurzzeitspeicher, einem "Flash-Speicher". Dieser Flash-Speicher kann auf verschiedene Weise aktualisiert werden, einschließlich einer Kommunikation durch eine geeignete Schnittstelle, wie einer parallelen oder seriellen Schnittstelle, oder einem Netzwerkadapter, sobald sich das Terminal in einem vorbestimmten Status befindet.

Es gehört deshalb zum Vorteil der vorliegenden Erfindung, ein Terminal verfügbar zu machen, das Information darstellen kann, die von einem Anwendungs-Server durch Ausführung eines populären Anwendungsprogramms unter der Microsoft Windows® NT-Betriebssystem-Umgebung erzeugt wurde.

Weiterhin gehört dazu, ein Terminal-Konfigurationssystem mit einer graphischen Benutzerschnittstelle für die Konfiguration der Terminal-Betriebsparameter zu schaffen.

Außerdem gehört dazu, ein Terminal für die Darstellung von Anwendungsprogramm-Information vorzusehen, welches einen Clienten-Personalcomputer in einer Client-Server-Netzwerk-Umgebung emuliert.

Ferner wird ein Terminal verfügbar, welches wünschenswerte Merkmale, die normalerweise mit einer Multiuser-Computer-Umgebung einhergehen, mit den wünschenswerten Merkmalen kombiniert, die normalerweise mit einer Client-Server-Umgebung einhergehen.

Es gehört zur Erfindung, über ein Terminal-Betriebssystem zu verfügen, welches Kommunikationen mit einer Fenster-Betriebssystem-Umgebung ermöglicht, während gleichzeitig ein vom PC/AT BIOS und PC DOS abweichender Standard benutzt wird.

Zur Erfindung gehören aber auch eine Vorrichtung und ein Verfahren für eine schnelle und geeignete Aktualisierung des Terminal-Betriebssystems.

Schließlich sollen mehrere residente Kenndaten für ein Terminal verfügbar sein, und zwar durch schnelles Umschalten zwischen den verschiedenen Kenndaten, ohne daß Erweiterungskarten ausgetauscht werden oder das Terminal vom Stromnetz genommen werden muß.

Weitere Einzelheiten, Merkmale und Vorteile der vorliegenden Erfindung ergeben sich aus der folgenden Beschreibung der Zeichnungen.

Es zeigen:

Fig. 1 eine schematische Darstellung einer allgemeinen Anordnung eines Anwendungs-Servers und eines Terminals gemäß vorliegender Erfindung,

Fig. 2 ein Blockschaltbild einer Architektur des neuen Systems,

Fig. 3 ein Blockschaltbild des Aufbaus des Steuer ASIC in Fig. 2,

Fig. 4 ein Flußdiagramm zum Ablauf der Schritte in einem Terminal,

Fig. 5 ein vereinfachtes Blockschaltbild der Setup-Schnittstelle zwischen einer GUI-Maschine und dem Rest des Systems,

Fig. 6 ein Blockschaltbild des Ablaufs eines Verbindungsaufbaus zwischen Terminal und Anwendungs-Server,

Fig. 7A einen Bildschirmaufbau gemäß vorliegender Erfindung und
Fig. 7B1—7B3 Datenstrukturen.

In Fig. 1 ist ein vereinfachtes System gemäß dem Prinzip der vorliegenden Erfindung dargestellt. Ein einzelner Anwendungs-Server 10 steht bidirektional mit einem oder mehreren Terminals 12 über ein geeignetes Netzwerk oder eine andere Kommunikationsleitung 14 in Verbindung. Diese Kommunikationsleitung kann aus einer RS232-Leitung, einem Wechselstromleitungsmodem oder einer Ethernetverbindung, beispielsweise einem verdrehten Leitungspaar oder einem Koaxkabel oder einer anderen geeigneten Verbindung, bestehen, beispielsweise einer Glasfaseroptik. In einer im Probetrieb zufriedenstellenden, hier beispielhaft erwähnten Anordnung wird der Anwendungs-Server mit dem WINDOWS® NT-Betriebssystem mit geeigneten Erweiterungen, z. B. dem von CITRIX angebotenen Winframe® OS, betrieben. Das CITRIX Fern-WINDOWS-Protokoll oder Erweiterungen schließen sowohl das ICA 3.0-Protokoll als auch Verbesserungen ein, welche für eine echte Multiuser-Fähigkeit innerhalb der WINDOWS® NT-Umgebung sorgen. Bei einer solchen Konfiguration kann es sich beim Anwendungs-Server 10 um einen auf einem INTEL Pentium® oder einem '486-Prozessor oder andere ähnliche Prozessoren, wie etwa ein DEC Alpha oder MIPS Prozessor, oder einer Zusammenschaltung mehrerer Prozessoren in Verbindung mit einer geeigneten RAM-Kapazität, basierenden Personalcomputer handeln. In einem Ausführungsbeispiel hat der Server 10 ein sechzehn Megabyte großes RAM für Winframe® OS, zuzüglich 1—8 Megabyte RAM pro gleichzeitig laufendem Benutzer, und zwar in Abhängigkeit davon, welche Anwendung für den einzelnen Benutzer abläuft.

In anderen geeigneten Ausführungsbeispielen kann der Anwendungs-Server 10 auch mit anderen Servern kommunizieren, so mit einem NetWare Fileserver 16, einem UNIX®-Host 18, anderen Personalcomputern 20 oder auch einem Internet Gateway 22. Über andere Verbindungen, wie einem Router oder anderen Kommunikations-Servern 24 kann der Anwendungs-Server 10 auch mit dahinter arbeitenden Rechnern 26 kommunizieren oder über andere Vorrichtungen auch mit anderen entfernten anwählbaren Benutzern 28.

Nähere Einzelheiten der Hardware-Architektur eines Terminals ergeben sich aus Fig. 2. Eine CPU 100, typischerweise ein Prozessor der X86-Familie oder wie in einem Ausführungsbeispiel ein 80386CXSA oder ein '486SXLC-Mikroprozessor, empfängt den Takt und Reset-Signale von einer logischen Takt-und-Reset-Schaltung 102. Die CPU 100 kommuniziert mit den anderen logischen Schaltungen über einen Adreßbus 105, einen Datenbus 106 und einen CTRL-Bus 108. Es ist wichtig, daß diese Bus-Verbindungen typischerweise weder mit dem IBM PC/AT-Standard noch mit einem anderen PC-Standard kompatibel sind, da mit der Erfindung beabsichtigt ist, viele, wenn nicht gar alle Mängel zu vermeiden, die sonst bei Personalcomputern dann auftreten, wenn sie in einer Netzwerkumgebung arbeiten.

Insbesondere verläuft der Adreßbus 105 von der CPU 100 sowohl zu einem Steuerungs-ASIC 110 (anwenderspezifischer IC 110) als auch zu einem Flash-Speicherbereich 112 und einem VGA-Kontroller 114. Der Datenbus 106 kommuniziert gleichermaßen mit dem Steuerungs-ASIC 110, dem VGA-Kontroller 114 sowie einer Speicheranordnung 116. Der CTRL-Bus 108 legt in ähnlicher Weise Steuersignale an den ASIC 110 und den VGA-Kontroller 114 an. Verschiedene logische Schaltungen 118 legen CONFIG, IDCS und DIAGCS an den ASIC 110 an.

Der ASIC 110 kommuniziert mit der Speicheranordnung 116 über einen DMA Bus 120 und ferner mit dem Flash-Speicher 112 über einen PD-Bus 122. Der PD-Bus 122 sorgt außerdem für eine Kommunikation zwischen dem ASIC 110 und dem Flash-Speicher 112, einem seriellen I/O und einem parallelen Kontroller 124, einem Tastatur- und Maus-Kontroller 126 sowie einem LAN- (= Local Area Network) Kontroller 128. Zusätzlich legt der ASIC 110 ein FCS-Signal (Prüfsignal) an den Flash-Speicher 112 an, sorgt für RAS-, CAS- und WE-Signale für die Speicheranordnung 116 und für COM1, COM2 und PPCS-Signale für den SIO-Kontroller 124. Weiterhin legt der ASIC 110 ein KBCS-Signal an den Tastatur/Maus-Kontroller 126 an und legt ein NETCS-Signal an den LAN-Kontroller 128. Schließlich gibt der ASIC 110 ein Lautsprecher-Freigabesignal SPEN an einen Lautsprecher 130.

Die CPU 100 kommuniziert auch mit dem SIO-Kontroller 124, dem Tastatur/Maus-Kontroller 126 und dem LAN-Kontroller 128 über einen Teil des Busses 105. Zusätzlich stellt der ASIC 110 dem SIO-Kontroller 124 Hardware-Interrupts INTA, INTB, INTP und dem KB/MS-Kontroller 126 Hardware-Interrupts KBINT und MSINT zur Verfügung und dem LAN-Kontroller 128 einen Hardware-Interrupt NINT.

Der Speicherbereich 116 besteht typischerweise aus einem DRAM-Speicher, obwohl andere Speicherarten in abgewandelten Ausführungsformen gleichfalls eingesetzt werden können. Jedoch liegt die Größe des DRAM im Bereich 116, der für den Betrieb des Terminals erforderlich ist, im Unterschied zum modernen Personalcomputer typischerweise im Bereich von 512 Kilobyte bis vier Megabyte. In einer beispielhaften Ausführungsform werden nur 23 Adreßleitungen und eine Byte-Auswahlleitung benutzt, wodurch die Speicherkapazität auf 16 Megabyte begrenzt werden kann. In anderen Ausführungsformen können unterschiedliche Speichergrößen eingesetzt werden.

Der Steuerungs-ASIC 110 weist in einem Ausführungsbeispiel Funktionsblöcke für die Bussteuerung, für die DRAM-Steuerung (hier typischerweise im Fast-Page-Modus mit gegenseitiger Versetzung), für einen Systemzeitgeber, für einen Lautsprecherzeitgeber und einen I/O-Kontroller. Der Steuerungs-ASIC kann außerdem als ein Gate-Array oder anderer hochintegrierter Schaltkreis implementiert sein, was näher im Detail in Verbindung mit Fig. 3 nachstehend beschrieben werden wird.

In einem Anwendungsbeispiel hat es sich als ausreichend erwiesen, den Flash-Speicher 112 in der Größenordnung von 512 Kilobyte zu halten; jedoch andere Anwendungen könnten eine Größe bis hinauf zu fünf Megabyte oder mehr erfordern. In dem bevorzugten Ausführungsbeispiel ist zwar der Bereich 112 als Flash-Speicher dargestellt, in anderen Ausführungsbeispielen könnte, jedoch ohne die Funktionalität der vorliegenden Erfindung zu verlassen, dieser Bereich auch aus einem EPROM oder einem SRAM oder anderen vergleichbaren Speichereinheiten bestehen.

Der SIO-Kontroller 124 kommuniziert mit COM1-, COM2- und Drucker-(oder Parallel-) Ports 132, 134 bzw.

136. Der SIO/Parallel-Kontroller 124 kann ein handelsüblicher 16 552-Baustein sein, so wie er von der Firma Startech angeboten wird. Der Tastatur/Maus-Kontroller 126 kommuniziert in ähnlicher Weise mit einer Tastatur 138 und einer Maus 140, während der LAN-Kontroller, welcher nicht in allen Ausführungsformen benötigt wird, mit einer LAN-Schnittstelle 142 kommuniziert. In einem Prototyp bestand der Tastatur/Maus-Kontroller 126 aus einem Standard-Tastatur-Kontroller, während die Tastatur und die Maus dem PS/2-Standard entsprechen, obwohl zumindest in einigen Ausführungsformen eine Modifizierung des Tastatur-Kontrollers möglich ist, um mit einer Vierdraht-Tastatur-Schnittstelle kompatibel zu sein, wie sie in der US-Patentschrift 4 706 068 beschrieben ist. Schließlich kann der LAN-Kontroller aus einem beliebigen geeigneten Netzwerk-Schnittstellen-Kontroller bestehen und jedem möglichen Netzwerk-Standard entsprechen, einschließlich 10BaseT, 10Base2 oder anderes. Die Netzwerkschnittstelle kann einen 512 Kilobyte-Speicher oder einen größeren für eine zusätzliche Codespeicherung aufweisen.

Der Video- und Graphik-Kontroller 114 ist funktional mit einem zweiten Speicherbereich 144 verbunden, um Video- und graphische Informationen zu speichern, die an einen Monitor 146 über ein MPS 148 geleitet werden. Der Video-Kontroller 114 kann beispielsweise ein Cirrus 5429-Baustein mit internem RAMDAC sein und kann einen Videospeicher in der Größenordnung von einem Megabyte haben, um hochauflösende Graphikdarstellungen zu ermöglichen, wie sie zumindest dem Video Graphics Array-Standard entsprechen. Es wird jedem Fachmann deutlich werden, daß die gesamten Speicheranforderungen entscheidend kleiner sind, als sie für einen Personalcomputer mit vergleichbarer Kapazität nötig sind, um WINDOWS®-Anwendungsprogramme ablaufen zu lassen. Ein Ruhesignal 150 kann von dem ASIC 110 bereitgestellt werden. Verschiedene Methoden zur Implementierung eines Ruhesignals können benutzt werden. Beispielsweise kann in einer Schwarz-Weiß-Version der vorliegenden Erfindung ein Signal an die Stromeinspeisung für den Monitor bereitgestellt werden, welches das Videosignal abschaltet und den Stromverbrauch, der für den Betrieb des Monitors notwendig ist, absenkt. In einer Farbversion kann das Synchronisationssignal in Übereinstimmung mit dem VESA-Standard manipuliert werden, um den Monitor leistungsmäßig herunterzuschalten. Es wird für den Fachmann deutlich, daß viele Funktionen, einschließlich der CPU, in einem oder mehreren integrierten VLSI-Bauteilen, wie einem ASIC, Gate-Arrays oder anderen Bauteilen, zusammengefaßt werden können.

Es ist ein besonderes Merkmal der Hardware gemäß der vorliegenden Erfindung, daß das Terminal-Betriebssystem, welches in dem Flash-Speicher 112 abgelegt ist, durch eine Vielzahl von Methoden aktualisiert werden kann, einschließlich der Kommunikation über eine geeignete Schnittstelle, wie die Parallel-Schnittstelle 136 oder den seriellen Schnittstellen 132 oder 134 oder einen Netzwerkadapter, wie die LAN-Schnittstelle 142. In einer bevorzugten Ausführungsform kann der Flash-Speicher durch eine Kommunikation mit dem Host-System aktualisiert werden, wenn sich das Terminal in einem vorherbestimmten Zustand befindet, wie z. B. durch Einstecken eines Rückführungsschleifensteckers, einer geeigneten Schlüsselsequenz oder anderen geeigneten Mitteln. In einer derartigen Anordnung erfolgt das Herabladen in das Speichersystem des Terminals, während die Kommunikation mit dem Host-Rechner weiterhin erlaubt ist. Der Host-Rechner stellt dann über die Kommunikationsverbindung aktualisierte Betriebsmerkmale entweder dem DRAM 116 des Terminals oder direkt dem Flash-Speicher 112 bereit. Danach wird, erforderlichenfalls, die aktualisierte Betriebssystem-Information im Flash-Speicher 112 gespeichert und das Terminal kehrt in den normalen Betriebszustand zurück, in dem das Herunterladen abgeschaltet ist.

Unter normalen Umständen startet ein System gemäß Fig. 2 den Betrieb nach einem Reset durch Beginn der Ausführung des Boot-Codes, der im Flash-Speicherbereich 112 abgelegt ist. Der Flash-Speicherbereich 112 kann in zwei Speicherbereiche angeordnet sein, wobei auf beide an einer vorbestimmten Adresse des CPU-Speicherplatzes zugegriffen wird, beispielsweise C00000—DFFFFF und E00000—FFFFFF, während der Rest des Speicherplatzes dafür vorgesehen ist, viele der sich aus Tabelle 1 ergebenden normalen PC-Funktionen bereitzustellen. Es wird deutlich, daß in einer bevorzugten Ausführungsform die Video/Graphik-Speicheradressen um acht Megabyte über die normalen PC-Speicheradressen hinaus verschoben sind. In dieser Ausführungsform kann der VGA-Chip in einem "kompatiblen" Modus benutzt werden (d. h. der lineare Adressenmodus ist ausgeschaltet), so daß der VGA-Chip nur auf einen Datenübertragungsblock-Puffer bei A0000h—AFFFFh (im Graphikmodus) oder bei B0000h—B7FFFh (für den MGA-Modus) oder bei B8000h—B8FFFh (für den CGA-Modus) anspricht, und daß diese Adressen auf 8A0000h—8AFFFFh verschoben werden usw. Die Adressen werden verschoben, um dem Speicherbereich 0 DRAM zu ermöglichen, zusammenhängend zu bleiben, während zur gleichen Zeit (und wie nachstehend näher erläutert wird) eine Emulation der normalen PC-Funktionen ermöglicht wird, die ausreicht, um eine WINDOWS®-Anwendungsprogramm-Information darzustellen.

Tabelle 1

5	FLASH Speicherbereich 0 Boot Block Flash	FFFFFF E00000
10	FLASH Speicherbereich 1 Datei-system Flash	DFFFFFF C00000
15	Netzwerk Karte	BFFFFFF B80000
20	Zusätzlicher Flash Speicher	B7FFFF B00000
25	Video/Graphik Kontroller	AFFFFFF 800000
30	DRAM Speicherbereich 1	7FFFFFF 400000
	DRAM Speicherbereich 0	3FFFFFF 000000

35 Speicherbereich 0 kann in einen 16 Kilobyte Boot Block im oberen Adressenbereich, beispielsweise bei FFC000H—FFFFFFH, angeordnet sein, gefolgt von zwei 8 Kilobyte Parameter-Blocks und weiter gefolgt von einer Vielzahl von Hauptblöcken bis hinunter zur Adresse F80000H (für eine 512 Kilobyte-Auslegung) oder hinunter bis zu F00000H (wenn eine 1 Megabyte-Auslegung vorliegt) oder bis E00000H (wenn eine 2 Megabyte-Auslegung vorliegt). Die Hauptblöcke des Speichers sind typischerweise als ein File System (Dateisystem) eingerichtet. Der Speicherbereich 1 des Flash-Speichers ist typischerweise einer File-System-Einrichtung zugeordnet und kann von den Adressen DFFFFFFH bis hinab zu C00000H reichen.

40 Nachstehend wird anhand von Fig. 3 der ASIC 110 näher erläutert. Ein Taktpuffer 160 empfängt ein CLK50-Signal und legt Taktsignale an die logische RESET SYNC Schaltung 162 an, an die logische DRAM-Steuerschaltung 164 und an die logische Zeitsteuerschaltung 166. Ein (INT)-Puffer 168 empfängt INT, INTA, INTB, MSINT, KBINT und NINT-Signale. Der INT-Puffer 168 legt eine Vielzahl von Signalen an die logische Interrupt-Steuerschaltung 170, welche ein Steuersignal vom CPU-Steuerungseingangspuffer 172 und ein Zeitgebersteuersignal von der logischen Zeitgebersteuerschaltung 166 empfängt und erzeugt ein INTR-Ausgangssignal, welches an die CPU angelegt wird. Der CPU-Steuerungseingangspuffer 172 legt ferner ein Steuersignal an die logische Cycle-Steuerungsschaltung 174, die ihrerseits ein Steuersignal an die DRAM-Steuerung 164 anlegt. 45 Die DRAM-Steuerschaltung 164 empfängt ein Aktualisiersignal von der Zeitgebersteuerungsschaltung 166; die Zeitgebersteuerungsschaltung 166 erzeugt außerdem das Freigabesignal SPEN für den Lautsprecher.

50 In einer bevorzugten Ausführungsform ist die logische Interrupt-Steuerschaltung 170 nicht mit dem Baustein 8259 kompatibel; gleichermaßen ist der Systemzeitgeber oder die Zeitgebersteuerungsschaltung 166 nicht mit dem Baustein 8254 kompatibel und arbeitet mit einer höheren Frequenz als ein 8254-kompatibler Baustein. Als Ergebnis werden einige der von der logischen Zeitgebersteuerungsschaltung gesendeten Interrupts höherer Frequenz im Kernel maskiert, aber andere werden durchgelassen, um im Durchschnitt die normale Zeit zwischen Interrupts anzunähern. Die höhere Frequenz ermöglicht somit die Emulation von Standard-PC-Funktionalitäten, sogar, obwohl die Zeit zwischen auf niedriger (Standard) -Frequenz emulierten Interrupts nicht gleichförmig sein kann.

60 Der ASIC 110 umfaßt auch einen CPU-Adreß-Eingangspuffer 176, welcher BE0, BE1 und A1-A23-Signale empfängt und legt ein Ausgangssignal an einen DRAM-Speicher-Adressen-Multiplexer "Mux 178" und zugleich an eine logische Mem/IO-Chip-Auswahl-Steuerschaltung 180. Die logische Mem/IO-Chip-Auswahl-Steuerschaltung 180 produziert eine Vielzahl von Ausgangssignalen einschließlich FLASHCS0 (FCS0)- und FLASHCS1 (FCS1)-Signale, und andere, die sich aus der Zeichnung ergeben. Ferner empfängt der ASIC 110 65 Signale DO-15 vom Datenbus 106 und legt sie an den CPU-Dateneingangspuffer 182 an. Der Puffer 182 legt Daten an einen Bdata-Ausgangspuffer und Zwischenspeicher 184 an, welcher ein Ausgangssignal BDO-15 erzeugt. Der Puffer 182 legt außerdem Daten an eine Stromversorgungssteuerschaltung 186 an, welche SLEEP und PWRDWN-Signale erzeugt.

Die Signale BDO-15 können ebenso Daten an den Bdata-Eingangspuffer 188 anlegen, der seinerseits diese Daten an den CPU-Datenausgangspuffer 190 leitet. Die logische Interrupt-Steuerschaltung 170 legt auch Signale an den Puffer 190 an. Ein Konfigurationsregister 192 stellt ein Konfigurationssignal bereit (beispielsweise ein solches zur Hardware-Konfiguration z. B. wieviel Flash-Speicher und/oder DRAM-Speicher benötigt wird, oder eine Konfiguration von der Monitor-Stromversorgung oder von einer Steckkarte, z. B. einer Netzwerkkarte), welches an den CPU-Datenausgangspuffer angelegt wird, der Daten an den Bus 106 zur CPU anlegen kann.

Unter Bezugnahme auf Fig. 4 können die Schlüsselemente des Terminal-Betriebssystems besser verstanden werden. Aus dem Vorhergehenden geht hervor, daß die Hardware gemäß der vorliegenden Erfindung mit einer üblichen Standard AT-Bus-Auslegung nicht kompatibel ist. Statt dessen stützt sich die vorliegende Erfindung auf Firmware ab, um die erforderlichen BIOS-Dienste für die oberen Software-Schichten vorzubereiten. In einer bevorzugten Ausführungsform ist die Firmware ausgelegt, um im virtuellen 8086-Modus abzulaufen, und zwar mit AT-kompatiblen Hardware-Komponenten, wie den Interrupt-Kontrollern und Zeitgebern, die in Software so dicht als möglich emuliert sind. Während in einer Ausführungsform ein Standard-Tastatur-Kontroller benutzt wird, würde in dem Fall, in dem ein Nichtstandard-Kontroller benutzt wird, die Schnittstelle zu einer solchen Einrichtung zusätzlich emuliert werden. Signale, z. B. Ein- und Ausgangssignale zu und an die Ports derartiger Hardware-Komponenten, werden abgefangen, um die Emulation zu ermöglichen. Des weiteren könnten die Speichermanagementmerkmale des Prozessors unter der Steuerung eines emulierten A20-Gates freigegeben werden, um den Bildumlauf zu simulieren, welcher normalerweise in einer Hardware bei einem Megabyte auftritt.

Gemäß Fig. 4 beginnt das Terminal-Betriebssystem mit der Abarbeitung eines Boot Blocks 300, gefolgt vom Laden eines Kernels 305. Der Kernel 305 besorgt viele der Abfang- und Wiedereinteilungsfunktionen der vorliegenden Erfindung, wie sie nachstehend näher erläutert werden. Nach Beendigung des Kernels 305 wird der IO.SYS-Code 310 geladen. Als nächstes wird der COMMAND.COM-Code 315 geladen, gefolgt von in einem AUTOEXEC.BAT-File abgelegten Ausführungsbefehlen. Das AUTOEXEC.BAT-File kann beispielsweise Tastatur- und Maustreiber umfassen, obwohl diese beiden Treiber nicht in jedem Zustand benötigt werden, sowie einen VGA XMS-Treiber. Es kann außerdem andere optionale Codes einschließen, wie das Auslösen einer Selbsttestsequenz, welche abläuft, wenn geeignete Bedingungen bestehen. In einem bevorzugten Ausführungsbeispiel ist ein Rückführungsschleifenstecker in einem Kommunikationsport eingesteckt, der bewirkt, daß die Selbsttestsequenz abläuft.

Der EXEC.COM-Code 325 wird dann geladen. An diesem Punkt und in Abhängigkeit von der Implementation wird das System entweder in den Setup-Modus übergehen oder Befehle des Anwenders können entweder in einen Eintritt in den Setup-Modus oder das Laden eines Netzwerk-Verbindungscode bewirken. In einem Ausführungsbeispiel tritt das System in einen Setup-Modus ein, um aktuelle Konfigurationsdaten zu erhalten, und setzt mit dem Laden des Netzwerk-Verbindungscode fort.

Wenn die Implementation es dem Benutzer erlaubt auszuwählen, und wenn der Setup-Modus durch den Anwender gewählt worden ist, zweigt die EXEC.COM 325 ab, um den SETUP oder GUI 330 ablaufen zu lassen. Wenn der Setup-Modus nicht aktiviert worden ist, wirkt die EXEC.COM 325 mit, daß Netzwerk-Treiber bei 335 auf- und abgeladen werden, und beginnt mit der Abarbeitung des Netzwerk-Verbindungscode 340 (wiederum ICA, verdrahter Draht, com oder anderes Netzwerk). In einer bevorzugten Ausführungsform enthält der Netzwerk-Verbindungscode eine wesentlich modifizierte Version des WinFrame für DOS-Clienten, deren Standard-Version bei Citrix Systems, Inc. verfügbar ist.

Unter Bezugnahme auf Fig. 5 läßt sich die Zusammenarbeit des Terminal-Betriebssystems und der Hardware-Architektur besser verstehen. Die in Fig. 5 dargestellte unterste Schicht ist das Eingangs/Ausgangssystem und die Hardware-Schicht 400. Die nächsthöhere Schicht ist die Schicht der Treiber 402, während die oberste Schicht die Anwendungsschicht 404 ist.

Beim Einschalten werden die Power-Up- und Init-Tests 406 in der Hardware-Schicht als Teil des Boot Blocks 300 ausgeführt. Die Power-Up- und Init-Tests 406 werden teilweise vom Flash-Speichersystem 112 heraus und zum Teil vom RAM 116 aus ausgeführt. Wenn die Einschalt-Selbsttests beendet sind, setzt das Terminal mit der Boot-Sequenz fort, die allgemein oben in Verbindung mit Fig. 4 beschrieben wurde, einschließlich des Rests im Boot Block 300, einer AUTOEXEC-Sequenz 408 und der COMMAND.COM-Sequenz, wie sie zuvor mit 315 bezeichnet wurde. Beides, die AUTOEXEC- und die COMMAND.COM-Datei, sind im Flash-Speicher abgelegt.

Nachdem die COMMAND.COM-Sequenz des Terminals ausgeführt wird, bewirkt dies, daß die AUTOEXEC-Datei geladen wird. Die AUTOEXEC ihrerseits bewirkt, daß die EXEC.COM 325 geladen wird. Wie oben bereits bemerkt, kann die EXEC.COM-Sequenz 325 entweder in ein Setup-Modul 330 oder in ein Netzwerk-Verbindungsmodul 340 verzweigen. Bei der anfänglichen Installation oder zu jeder Zeit danach, wenn Betriebsparameter des Terminals eine Variation oder Änderung verlangen, läßt man das Setup-Modul 330 ablaufen. Das Setup-Modul 330 empfängt Informationen von einem oder mehreren Setup-Datenfiles 418 und setzt die GUI-Maschine 420 in Gang. Die GUI-Maschine 420 ihrerseits kommuniziert mit einem Tastaturreiber 422, einem Maustreiber 424 und den Dateien- und Speichertreiber 426 des Terminal-Betriebssystems. Zusätzlich kommuniziert die GUI-Maschine 420 gleichfalls mit dem Videoeingangs- und -ausgangssystem 428, welches seinerseits Daten an den Video-Kontroller 430 anlegt, der beispielsweise auf einem Cirrus 5429 Graphik-Prozessor basieren kann, um eine Videoanzeige während der Setup-Sequenz zu erzeugen. Die Setup-Sequenz wird detaillierter unter Bezugnahme auf Fig. 5 beschrieben.

Der Tastaturreiber 422 seinerseits kommuniziert mit der Tastatur-Kontroller-Hardware 432, welche beispielsweise ein konventionelles PS/2-Tastatur-Ein/Ausgabesystem, eine universelle serielle Busschnittstelle und zumindest in einigen Ausführungen auch eine Vierdraht-Tastatur-Schnittstelle sein kann, wie sie in der zuvor schon erwähnten US-Patentschrift 4 706 068 beschrieben ist. In gleicher Weise kommuniziert der Maustreiber 424 typischerweise auch zu geeigneten Zeiten mit einem Maus-Ein- und Ausgabesystem 434. Während all dieser

Operationen werden das Flash-File und die Speicher-Dienst-Teile 426 des Terminal-Betriebssystems typischerweise aus dem Flash- und RAM-Speicher herausgeführt.

Wie deutlicher in Verbindung mit Fig. 5 ausgeführt, ermöglicht es der Setup-Ablauf dem Benutzer, die Konfigurationsinformation auf dem Terminal zu spezifizieren, einschließlich solcher Parameter, wie die Netzwerk-Schnittstelle und verwandte Konfigurationsdetails, die Sprache, Farbe oder andere Parameter. Wenn diese Parameter einmal spezifiziert sind, werden die Daten in den Verbindungsdatendateien 440 abgelegt.

An diesem Punkt ist der Benutzer so weit, das Terminal-Setup-Modul 414 zu verlassen und zur EXEC.COM zurückzukehren. Wenn ihm eine Fortsetzung erlaubt wird, kann der EXEC.COM-Vorgang 412 zu einer Abzweigung in das Netzwerk-Verbindungsmodul veranlaßt werden, 416. Das Netzwerk-Verbindungsmodul 340 beginnt durch Zurückholen der in der Verbindungsdatendatei 440 und der Befehlsleitung des Verbindungsmoduls abgelegten Daten, wodurch dem Applikationsserver mitgeteilt wird, wie mit Rest von Treiber und Hardware-Schichten des Terminals in Verbindung zu treten ist. Insbesondere kommuniziert das Netzwerk-Verbindungsmodul mit dem Tastaturreiber 422, dem Maustreiber 424, dem Video Ein/Ausgangssystem 428 und dem File und Speicher-Dienst-Teil 426 des Terminal-Betriebssystems. Zusätzlich verbindet das Netzwerk-Verbindungsmodul auch eine in hardware ausgeführte serielle Schnittstelle 442 und in einigen Ausführungen gleichfalls eine Hardware-Netzwerk-Schnittstelle 444. Die Netzwerktreiber 444 werden in einer Ausführungsform aus dem RAM 116 heraus ausgeführt, kann aber ebenso auch aus dem Flash-Speicher 112 heraus ausgeführt werden. Die serielle Schnittstelle 442 kann eine konventionelle RS232-Schnittstelle sein, aber auch eine andere Form einer seriellen Verbindung, wie beispielsweise der Universal Serial Bus oder USB.

Anhand der Fig. 6 wird die Arbeitsweise der GUI-Maschine 420, wie sie in Fig. 5 dargestellt ist und wie sie während des Setup-Moduls des Terminals 12 abläuft, näher beschrieben. Die GUI-Maschine arbeitet nur während des Setup-Modus und stellt eine rudimentäre, graphische Benutzer-Schnittstelle während der Konfigurationsoperation bereit.

Wie schon in Verbindung mit der Beschreibung von Fig. 5 erwähnt wurde, beginnt die Operation gemäß Fig. 6, wenn die Setup-Sequenz während der Boot-Phase des Terminals aufgerufen wird. Diese Setup-Sequenz kann durch eine Sequenz von Tastenanschlägen oder in anderer geeigneter Weise aufgerufen werden. Die Setup-Sequenz startet mit einem Aufruf des Setup-Codes 502, der seinerseits Informationen von der Setup-Data-Datei 418 bezieht. Die Setup-Data-Datei 418 identifiziert die verschiedenen Konfigurationsoptionen, die im Terminal verfügbar sind. Der Setup-Code 502 kommuniziert bidirektional mit einer RAM-Struktur 504 und bewirkt außerdem, daß bestehende Verbindungsinformationen von der Verbindungsdatendatei 440 in die RAM-Struktur 504 geschrieben werden. Die GUI-Maschine 420 kommuniziert außerdem bidirektional mit der RAM-Struktur, um aktuelle Informationen in einer nachstehend näher als Bereiche, Gruppen und Auswahlen beschriebenen Anordnung einzustellen und anzuzeigen. Zusätzlich stellt eine Hardware-Schnittstelle 506 Videoinformation an den Video-Kontroller 430 bereit, während sie auf die Informationen, die vom Benutzer über die Maus 260 oder die Tastatur 250 empfangen werden, reagiert.

Der Setup-Code erlaubt es dem Benutzer, eine Mehrzahl von Konfigurationsmenüs für die Bedienungseigenschaften des Terminals zu durchlaufen, wie z. B. die auf dem Terminal anzuzeigende Sprache, die Art der Netzwerkverbindung usw. Fig. 7A zeigt eine Darstellung eines Setup-Bildschirms, wie er im Konfigurationsmodus des Terminals benutzt wird. In einer bevorzugten Ausführungsform werden die Setup-Bildschirme graphisch dargestellt. Wenn der Benutzer diese Konfigurationsbildschirme durchläuft, können die Konfigurationsdaten selektiv durch den Anwender durch Benutzung der Tastatur und der Maus aktualisiert werden. Die aktualisierten Daten werden in der RAM-Struktur 504 beibehalten, bevor sie in die Verbindungsdatendateien 436 geschrieben werden. Jedoch können in einer gegenwärtig bevorzugten Ausführungsform gewisse Daten dynamisch aktualisiert werden, während andere Daten bis zur Beendigung der Setup-Sequenz nicht aktualisiert werden. Nach Beendigung der Setup-Sequenz einschließlich des Einschreibens von jeglichen verbleibenden Konfigurationsdaten in die Verbindungsdatendateien 436 endet die Setup-Sequenz und kehrt zur EXEC.COM 325 zur Einleitung des Netzwerkverbindungsmoduls 340 zurück, der in Fig. 5 gezeigt ist.

Wieder zurück zur Fig. 7A; dort wird das ganze Fenster, in welchem die Dateien erscheinen, hier als Bereich 600 bezeichnet. Innerhalb eines jeden Bereichs 600 sind eine oder mehrere Gruppen 610, und jede Gruppe 610 weist eine oder mehrere Auswahlen 620 auf. So umfaßt im Beispiel von Fig. 7A die "Kommunikations"-Gruppe die Wahlmöglichkeiten Serial Port, TCP/IP, SPX und IPX, von denen jede mit einer Region 630 verbunden ist, mit der angezeigt wird, daß diese Auswahl getroffen oder ausgewählt wurde.

Mit Bezug auf die Fig. 7B1—7B3 werden die mit der Konfigurations-Software verbundenen Datenstrukturen dargestellt. Insbesondere wird eine Liste von Bereichszeichern in der AREA_LIST 700 gefunden. Die Strukturen, auf die von der AREA_LIST gezeigt wird, umfassen Grenzen, Größe, Titel und Gruppen, wie sie für alle Bereiche zugeteilt sind, wie sie im Setup-Prozeß definiert wurden. Wie zuvor angemerkt, erscheint jeder Bereich als ein Fenster auf dem Schirm. Außerdem werden alle Bereiche, die gegenwärtig angezeigt werden, in der DISP AREA_LIST 702 angezeigt. In einer beispielhaften Ausführungsform ist der erste aufgelistete Bereich als der Bodenbereich angezeigt und der letzte angezeigte Bereich ist der oberste angezeigte Bereich. In einer beispielhaften Ausführungsform wird die Überlappung von Fenstern gestattet, obgleich ein Überlappen nicht notwendigerweise in allen Ausführungsformen erforderlich ist.

Mit 704 ist die Datenstruktur für GROUP_LIST bezeichnet, die alle Gruppen auflistet, die beim SETUP-Prozeß in allen Bereichen definiert wurden, die in der AREA_LIST 700 gefunden wurden. Wie zuvor erwähnt, enthält jeder Bereich eine oder mehrere Gruppen. Eine optionale Datenstruktur 706 für eine STRING_LIST kann ebenso vorgesehen werden, und eine FILE_LIST 708 wird als ein Verzeichnis für "bitmap"-Bilder bereitgestellt, welche in vielen Zuständen in den verschiedenen Bereichen, Gruppen und Auswahlen benutzt werden können.

Die Struktur der AREA_LIST ergibt sich aus 710. Sie schließt einen Block für einen Bereich ID 712, einen

Zeiger zum nächsten Bereich 714, einen Zeiger zum vorhergehenden Bereich 716 und einen Strukturzeiger 718 ein. Der Strukturzeiger 718, der zu jedem Bereich ID 712 gehört, zeigt auf eine Bereichsstruktur 715, welche den Bereich ID 712 zusammen mit einem ABS_X Eintrag 720 und einem ABS_Y Eintrag 722 einschließt, um die Position dieses Bereichs in Bezug auf die obere linke Ecke des Bildschirms anzugeben (in einer bevorzugten Ausführungsform der Erfindung). Die Bereichsstruktur 715 schließt ferner einen ROWS Eintrag 724 und einen COLUMNS Eintrag 726 ein, die zusammen die Größe des Bereichs spezifizieren. Ein FLAGS Eintrag 728 legt fest, ob sich eine Umrahmung um den Bereich erstreckt. Ein TITLE POSITION Eintrag 730 und ein TITLE BAR Eintrag 732 spezifizieren den Text des Titels und dessen Anordnung innerhalb der Titelzeile des betreffenden Bereichs, während ein MAX_STR_LEN Eintrag 734 die maximale Anzahl von Zeichen für den Titel spezifiziert.

Zusätzlich enthält die Bereichsstruktur 715 einen Eintrag 736 für die Anzahl von Gruppen innerhalb eines bestimmten Bereichs. Ein AREA_MPTR Eintrag 738 spezifiziert den "Hot Spot" des Maus-Zeigers innerhalb des Bereichs, während ein Eintrag DEF_BUTTON 740 festlegt, welcher Knopf innerhalb des Bereichs den Standard (Default) vorgibt. Der Standard-Knopf wird aktiviert, wenn die "Enter"-Taste gedrückt wird. Ein CAN_BUTTON Eintrag 742 spezifiziert den "Cancel"-Knopf, der aktiviert wird, wenn die "ESC"-Taste gedrückt wird. Schließlich ist eine Liste von Zeigern bei 744A-744N spezifiziert, einer für jede Gruppe, die zu dem Bereich gehört. Jeder Gruppenzeiger 744 zeigt zu einem zugehörigen Gruppen-Strukturblock 764, wie nachstehend ausgeführt. Eine "Hot Key"-Liste kann ebenfalls für den Bereich definiert werden.

Die bei 748 gezeigte Struktur von DISP_AREA_LIST ist im wesentlichen mit der Struktur von AREA_LIST 700 identisch und enthält Blöcke für den Bereich ID, den nächsten Bereich, den vorhergehenden Bereich und Strukturzeiger. Wie bei AREA_LIST 700 zeigt auch die DISP_AREA_LIST 748 auf die Bereichsstruktur 715. Eine ähnliche Struktur ist für die GROUP_LIST 704 bei 750 gezeigt und sie enthält eine Gruppe ID 752, einen Zeiger auf die nächste Gruppe 754, einen Zeiger auf eine vorhergehende Gruppe 756 und einen Gruppenstrukturzeiger 758. Eine ähnliche Struktur kann auch für die optionale STRING_LIST 706 vorgesehen werden und sie kann eine Zeichenfolge (String) ID 760, einen Zeiger auf die nächste Zeichenfolge 762, einen Zeiger auf die vorhergehende Zeichenfolge 764 und einen Zeichenfolge-Strukturzeiger 766 aufweisen.

Es wird noch einmal auf den Gruppen-Strukturzeiger 758 zurückgekommen. Dieser zeigt auf den Gruppen-Struktur Block 746 und enthält die Gruppen ID 752, einen PARENT_SELECT_ID Eintrag 780, um die Auswahl zu identifizieren, die, sobald sie aktiviert wird, automatisch diese Gruppe aufrollt, einen HOTSPOT_COUNT Eintrag 782, um die Anzahl von Maus-"Hot Spots" innerhalb der Gruppe zu identifizieren und ferner GSTART_X 784 und GSTART_Y 786 Einträge, um die relative Lage der Gruppe innerhalb des Bereichs zu bestimmen. In einer bevorzugten Ausführungsform werden beide, die Lage der Gruppe und der Auswahl, in Bezug auf die obere linke Ecke des sie enthaltenden Bereichs spezifiziert; aber auch andere Beziehungen, die akzeptabel erscheinen, können definiert werden, z. B. die Spezifizierung der Lage einer Auswahl relativ zur Lage ihrer Gruppe. Das wichtigste ist, daß sichergestellt ist, daß alle Merkmale eines Bereichs ihre Position innerhalb des Bereichs beibehalten, wenn der Bereich bewegt wird.

Der Gruppen-Struktur Block 746 umfaßt auch ROWS und COLUMNS Einträge 788 bzw. 790, um die Größe der Gruppe festzulegen, ebenso wie einen GFLAGS Eintrag 792, um die Umrandung der Gruppe festzulegen. Zusätzlich kann ein QUICK_KEY_POSITION Eintrag 794 und ein QUICK_KEY_STROKES Eintrag 796 für "Hot Key"-Kombinationen, die der Gruppe zugeordnet sind, spezifiziert werden.

Weiterhin und ähnlich wie für die Bereichsstruktur, können Einträge für Titelposition 798, Gruppen-Label 800 und MAX_STR_LEN 802 vorgesehen werden. Zusätzlich ist ein NUM_OF_SELECTS Eintrag 804 zur Identifizierung der Anzahl der Auswahlen vorgesehen, die in einer Gruppe enthalten sind. Als nächstes ist ein Eintrag 806 für AID_ATTACH als Rückreferenz zum Bereich ID 712 vorgesehen, zu dem die spezielle Gruppe gehört. Der AID_ATTACH Eintrag 806 ist nicht in allen Fällen erforderlich, aber verbessert zumindest in einigen Zuständen die Ausführung. Schließlich zeigt eine Liste von Zeiger-Einträgen 808A bis 808N jeweils auf eine ausgewählte Struktur, die mit der speziellen Gruppe verbunden ist. Wie nachstehend näher erläutert wird, kann eine Vielzahl von Auswahl-Strukturen mit jeder Gruppe verbunden werden, jedoch sind einige Elemente den verschiedenen Typen gemeinsam. Deshalb zeigt der erste Zeiger 808A auf einen SELECT_COMMON-Struktur-Block 810. Wiederum zurückkommend auf den Bereichs-Struktur Block 715 zeigen der Standard-Knopf Eintrag 740 und der "CANCEL"-Knopf Eintrag 742 ebenfalls auf den gemeinsamen Auswahl-Strukturblock 810.

Der SELECT_COMMON Struktur-Block 810 enthält einen Auswahl ID Eintrag 812, einen Eintrag 814, der eine Rückreferenz zur Gruppen ID ergibt, einen REL_X und REL_Y Eintrag 816 bzw. 818 zusammen mit ROWS und COLS Einträgen 820 bzw. 822, um die Lage und die Größe der Auswahl zu spezifizieren, einen QUICK_KEY_POS und einen QUICK_KEY_CHR Eintrag 824 bzw. 826, um die "Hot Key"-Kombinationen zu spezifizieren, die zu der Auswahl gehören, einen MAX_STR_LEN Eintrag 828 und eine Auswahl-Zeichenfolge 830, um die maximale Größe und den Titel der Auswahl zu spezifizieren, und einen SFLAGS Eintrag 832, um die Eigenschaften der Auswahl zu spezifizieren.

Zusätzlich wird auch ein SELECT_TYPE Eintrag 834 bereitgestellt. Wie zuvor erläutert, sind verschiedene Arten von Auswahlen verfügbar und insoweit wird auf Fig. 7A Bezug genommen. Die verschiedenen Arten von Auswahlen, die innerhalb einer Gruppe zur Verfügung stehen, hängen vom Datentyp ab, der für diesen Schritt der Konfiguration des Terminals erforderlich ist. In einigen Beispielen besteht die Auswahl lediglich im Drücken eines Knopfes (Knöpfe 640); in anderen Beispielen besteht die Auswahl in der Aktivierung und Deaktivierung eines Merkmals, wie eines Prüfblocks (siehe 650 in Fig. 7A); in anderen Beispielen muß eine von verschiedenen Auswahlen getroffen werden, wie es mit den Gruppen "Communication" 660 und "Serial Port" 670 in Fig. 7A dargestellt ist. Gemäß weiteren Beispielen kann ein Bild ausgewählt werden, während bei anderen spezieller Text ausgewählt werden muß. In einigen Beispielen müssen Felder ausgefüllt werden (680 in Fig. 7A), während in anderen eines von mehreren Feldern ausgefüllt werden muß. Obwohl diese Arten von Auswahlen in einer

beispielhaften Ausführungsform implementiert wurden, ist die Aufzählung keinesfalls als vollständig anzusehen und selbstverständlich können andere Auswahlen aus der hier gegebenen technischen Lehre leicht ebenfalls implementiert werden.

Für eine Auswahl "Ausfüllen" sind Cursor-Start und Cursor-Ende Einträge 836 bzw. 838 bereitgestellt, zusammen mit einem "First Displayed" Eintrag 840 zum Identifizieren, von welchem Zeichen an die Zeichenfolge angezeigt werden soll. Zusätzlich ist sowohl ein LABEL_REL_X Eintrag 842 als auch ein LABEL_REL_Y Eintrag 844 und ein LABEL_STR Eintrag 846 vorgesehen.

Für einen Auswahltyp "einer von vielen" sind Einträge von NUM_OF_SEL_ROWS 848 und NUM_OF_SEL_COLS 850 vorgesehen. Einträge sind auch für die Anzahl von Optionen 852 und die "default" Option 854 vorgesehen, ebenso ein Quick-Key-Pointer 856 und ein Flag-Pointer 858, um die Anzahl der Optionen zu identifizieren, die aktiv sind. Schließlich gibt es eine SEL_STR_SIZE 860.

Für einen "Bild"-Typ der Auswahl ist nur ein Eintrag für die Datei ID 708 nötig und ein Bildzeiger 862 muß spezifiziert werden.

Für einen "Felder"-Typ der Auswahl gibt es eine "Child-Group" ID Eintrag 864 zusammen mit einem Child-Group- (= Ablegergruppen) Zeiger, welcher auf eine Gruppenstruktur derjenigen Art zeigt, wie sie mit dem Gruppenstrukturblock 746 dargestellt ist. Die Ablegergruppe wird automatisch aufgerufen, sobald die Elternauswahl aktiviert ist und eine aus einer Gruppe von Feldern ausgewählt ist.

Für eine "List of Strings"- (= "Liste von Zeichenfolgen") Auswahl sind Einträge für die Anzahl der Optionen 868, die maximale Länge des Optionstitels (oder MAX_OP_LEN) 870, ein Eintrag 872 für den horizontalen Darstellungsversatz, ein Eintrag 874 für den vertikalen Darstellungsversatz, zusammen mit einer X-Label Position 878 und einer Y-Label Position 880 vorgesehen. Schließlich ist eine Label-Zeichenfolge 882 und ein Eintrag 884 für die Größe der Zeichenfolge der Auswahl vorgesehen.

Nochmals auf den AREA MPTR Eintrag 738 zurückkommend, wird der Maus-Zeiger-"Hot Spot" durch eine Struktur spezifiziert, die einen Bereichs-ID-Eintrag 900, einen Gruppen-ID-Eintrag 902 und eine Auswahl-ID 904 umfaßt. Zusätzlich ist ein Options-Auswahl-Typ 906 vorgesehen, der den Typ der Auswahl spezifiziert, mit der ein spezieller "Hot Spot" assoziiert ist. Ferner gibt es Rückreferenzen Einträge 908 und 910 für die Gruppen ID innerhalb des Bereichs und die Auswahl ID innerhalb der Gruppe. Weiterhin spezifizieren vier Einträge 912A-D sowohl die oberen linken X und Y Positionen, wie auch die unteren rechten X und Y Positionen für den Maus-Darstellungspunkt ("Hot Spot"), und zwar zusammen mit einem Eintrag 914 für ein Maus-Kennzeichen ("flag"), der bewirkt, daß der Mausdarstellungspunkt aktiviert wird, sobald das richtige Menü angezeigt wird. Zusätzlich zu den soeben beschriebenen Darstellungspunkten sind zusätzliche Darstellungspunkte am Anfang und am Ende jeder Anzeigenliste bereitgestellt, um ein Rollen zu ermöglichen, und ähnliches befindet sich in dem Titelleistenbereich eines Bereichs, damit das Bereichsfenster bewegt werden kann.

Zusätzlich zu den vorgenannten Strukturen wird auch eine Datenstruktur bereitgestellt, um die aktuell ausgewählten Einträge aus den verschiedenen Auswahlmöglichkeiten beizubehalten. Der aktuelle Daten-Struktur-Block ist bei 950 dargestellt und enthält einen Eintrag 952 für die Anzahl von Bereichen, die momentan vom SETUP definiert sind; ein Eintrag 954 gibt an, wieviele Bild-Dateien definiert sind; Einträge 956 bzw. 958 geben an, wieviele Gruppen und Auswahlen definiert wurden, ein Eintrag 960 dient der Zuordnung einer vorbestimmten Maximalanzahl von Auswahlen. In einer beispielhaften Ausführungsform ist die Maximalanzahl von Auswahlen in Zehnerblöcken zugeordnet.

Zusätzliche Einträge 962 und 964 sind für die Anzahl von Pixel pro Spalte bzw. Reihe vorgesehen, ebenso wie ein Eintrag für den Font 966, ein Bereichs-Focus-Eintrag 968, ein Gruppen-Focus-Eintrag 970 und ein Zeichenfolge-Focus-Eintrag 972. Weiterhin gibt es einen Maus-Focus-Eintrag 974 zur Spezifizierung des Darstellungspunktes. Weiterhin können OFOCUS und TFOCUS Einträge 976 bzw. 978 vorgesehen werden, um die ausgewählten Auswahl-Optionen und Auswahl-Typen mit einem Tastatur Fokus zu spezifizieren. Noch weiter gibt es IFOCUS und JFOCUS Einträge 980 bzw. 982 für die "Hot Spot"-Einträge 908 und 910 vom oben beschriebenen Maus-Struktur-Block. Schließlich spezifiziert ein Menü-Eintrag 986 die Identifizierung des aktuellen Menü-Focus, zusammen mit den Einträgen 988 und 990 zur Definition der Bereichs- und Gruppenumrahmungen, zusammen mit einem OFLAGS Eintrag zur Spezifizierung der Mausmodi.

Die den laufenden Zustand der Auswahlen spezifizierende Information befindet sich in einer ACTIVE-SELECT-Struktur 1000. Jede Struktur umfaßt einen Knopf-Eintrag 1002, einen STFLAGS- oder Auswahl-gemeinsame-FLAGS Eintrag 1004 und einen Aktiveintrag, welcher den aktuellen Status aller Auswahlen speichert, aus dem diese Daten für den SETUP Code verfügbar gemacht werden können.

In einer bevorzugten Ausführungsform kann auch eine Ereignis-Queue- (= Warteschlange) Struktur 1010 eingespeist werden, um Tastaturschläge oder Mausbewegungen in einer Ereignis-Warteschlange aufzuzeichnen.

Wie zuvor ausgeführt, ist es ein Schlüsselmerkmal der vorliegenden Erfindung, daß das Betriebssystem des Terminals der vorliegenden Erfindung nicht mit einem Standard PC/AT BIOS oder DOS kompatibel ist. Dennoch ist es erforderlich, daß das Terminal-Betriebssystem gewisse Funktionen unterstützt, um die Fähigkeit, Anwendungsdateien in einer Multiuserumgebung anzuzeigen, beizubehalten, wie beim Kommunizieren mit einem CITRIX-Clienten oder anderen unterstützten Emulationen. Es läßt sich eine Liste mit Standard IO.SYS und BIOS.SYS Funktionen aufstellen, welche von der vorliegenden Erfindung unterstützt werden; für den Fachmann ist deutlich, daß nicht alle Standard BIOS oder DOS Funktionen eingeschlossen werden müssen. Manche Funktionen werden nicht unterstützt. Andere Merkmale werden nur teilweise unterstützt. So wird die Funktion 36h (Get Disk Free Space) (= Ermittle freien Speicherplatz der Platte) deshalb nur teilweise unterstützt, weil ein Flashspeicher anstelle einer Harddisk (= Festplatte) benutzt wird. Ähnlich wird die Funktion 33h (Get/Set System Value) (= Ermittle/Setze Systemwert) insoweit in Abhängigkeit von der Funktion und dem Flag unterstützt, jedoch die "Control-Break" Funktion wird nicht unterstützt. In gleicher Weise werden die

Funktionen 2Ah bis 2Dh (die Get/Set Date/Time-, d. h. Ermittle/Setze Datum/Zeit, Funktionen) nur teilweise unterstützt, da keine Echtzeit-Hardware im Terminal der vorliegenden Erfindung vorhanden ist. Lediglich die "Get Time"- (= Ermittle Zeit) Funktion wird unterstützt, so daß sie zur Messung der Dauer von Ereignissen herangezogen werden kann, ohne die absolute Zeit wiederzugeben.

Zusätzlich ist das Flashfile-System in einer bevorzugten Ausführungsform der Erfindung in mehrere Einzelverzeichnis-Laufwerke partitioniert. Jedoch weist das Flashfile-System in Gegensatz zu konventionellen Diskfile-Systemen keine Einteilung in Cluster oder Sektoren auf. Dateien in jedem Laufwerk oder in jeder Partition wachsen von unten nach oben in der Partition, während die Directory-Einträge von oben nach unten wachsen. Daten werden fortlaufend ohne Fragmentierung gespeichert. Die Verzeichniseinträge, die in einer bevorzugten Ausführungsform sechzehn Bytes lang sind, entsprechen allgemein einem DOS-Verzeichniseintrag; jedoch werden Elemente, die normalerweise reserviert sein würden, definiert, um es zu ermöglichen, daß eine Datei aus dem Flash anstatt aus dem DRAM ausgeführt wird. Dies schließt die Startadresse der Datei im Flash ebenso ein, wie das Rückabbildungssegment der Datei im DOS Adressenraum.

Die Dateilöschung ist zwar der Löschung konventioneller DOS-Dateien weitgehend ähnlich, unterscheidet sich aber in wichtigen Details. Wenn eine Datei in der vorliegenden Erfindung gelöscht werden soll, wird das erste Byte im Verzeichnis-Eintrag auf 0 statt auf E5h gesetzt. Dieser Schritt erfolgt ohne Löschung eines Flashblocks. Nachfolgende Dateien werden dann in den nächsten verfügbaren Platz eingeschrieben. Falls jedoch für eine nachfolgende Datei nicht genügend Platz zur Verfügung steht, wird der Flashblock für die gelöschte Datei gelöscht und ungelöschte Dateien werden in den Flashblock dahin zurückgeschrieben, wo die gelöschte Datei beibehalten worden war. Wie schon zuvor erwähnt, ist eine Dateifragmentierung zumindest in einigen Ausführungsbeispielen nicht erlaubt.

Das Flashfile-System unterstützt konventionelle DIR, TYPE und DEL Befehle, unterstützt einen neuen "DEBUGMSG" Befehl zur Erzeugung einer DEBUG Mitteilung und unterstützt auch Programmausführungen durch Batchfiles. Das Dateisystem unterstützt auch die AUTOEXEC.BAT Datei und auch das Laden und Ausführen von *.EXE und *.COM Dateien und Int 21h und Int 27h. Jedoch unterstützt das Dateisystem in zumindest einigen Ausführungsformen nicht die CONFIG.SYS Datei oder .SYS-Gerätetreiber. Desgleichen unterstützt das Dateisystem nicht die Batchfile-Befehle (außer Programmausführung), I/O-Umleitungen, Pipes oder Interrupts 20h (Programm beenden), 22h (Adresse beenden), 23h (Ctrl-Break Adresse verlassen), 24h (Kritische Fehler Abwicklungs-Vektor), 25 h (Absolute Disk Read), 26h (Absolute Disk Write) und 2Fh (Multiplex Interrupt).

Aus dem Vorhergehenden wird deutlich, daß, während eine ausgewählte Gruppe der Standard BIOS und DOS Funktionen emuliert oder anderweitig vom Terminal-Betriebssystem der vorliegenden Erfindung unterstützt werden, eine sehr bedeutende Anzahl der Standard-BIOS- und DOS-Funktionen nicht unterstützt werden. Zusätzlich lassen sich sogar jene BIOS- und DOS-Funktionen, die unterstützt werden, nicht unter Standard AT kompatibler Hardware ausführen. Statt dessen richtet der Teil des Terminal-Betriebssystems, der in Fig. 4 als der "Bootblock" 300 und als "Kernel" 305 bezeichnet werden, die Fähigkeit zum Emulieren dieser Funktionen ein.

Die vom Bootblock 300 unterstützten Dienstfunktionen schließen ein: GET FLASH DRIVE SIZE, welcher den Flashspeicher 438 nach der Laufwerksgröße abfragt; READ FLASH DRIVE zum Lesen von Daten aus dem Flashspeicher 438; WRITE FLASH DRIVE zum Schreiben von Daten in den Flashspeicher 438; GET FLASH DRIVE BLOCK SIZE zum Abfragen der Blockgröße des Speichers 438; ERASE FLASH DRIVE BLOCK zum Löschen von Daten aus dem Speicher 438; WARM REBOOT wird beim Herstellertest benutzt, um wiederholt durch die Einschalt diagnose zu schleifen, aber nicht in normalen Betrieb benutzt; GET BOOT BLOCK DATE zur Abfrage der Daten des Bootblocks; CLEAR KEYBOARD CONTROLLER I/O BIT wird zur Steuerung von Komponenten benutzt, die mit dem Tastaturkontrolller verbunden sind; und schließlich SET KEYBOARD CONTROLLER I/O BIT zum Einstellen des Tastatur-Kontrollers.

Die Betriebsweise des Kernel 305 wird nun im Detail beschrieben. Der Kernel umfaßt im wesentlichen drei Dienstfunktionen. Die erste ist eine "ACTIVE VIDEO INT 10h" Funktion, welche die normalen int 10h Funktionen für Videodienste aktiviert. Der Interrupt int 10h wird anfänglich vom Kernel 305 abgefangen, um die Darstellung von Text-Modus-Mitteilungen von den verschiedenen Gerätetreibern zu unterdrücken, während diese geladen werden. Nachdem die Treiber alle geladen sind und das Terminal in den Graphikmodus gelangt, wird die ACTIVE VIDEO INT 10h Funktion aufgerufen, um die normale int 10h Operation zurückzusetzen. Zusätzlich schließt der Kernel 305 eine zweite Funktion ein, die "SET POWER DOWN TIME", welche verschiedene Stromsparfunktionen (wie "ENERGY STAR compliance") und eine Zeitverzögerung zum Aktivieren dieser Funktionen einstellen kann. Schließlich umfaßt der Kernel 305 eine "PROCESS DOS INTERRUPTS" Funktion, welche jedesmal vom Bootblock oder anderen Teilen des Kernels 305 aufgerufen werden kann, wenn es nötig ist, anstehende DOS Interrupts auszuführen, welche eine Echtzeit-Verarbeitung in einer DOS-Umgebung erfordern. Die am meisten auszuführenden Interrupts in dieser Funktion sind Maus- und Tastatur-Interrupts, obwohl auch Zeitgeber-, Seriell-, Parallel- und Netzwerk-Interrupts vorkommen. Das Einfangen durch diese Funktion verhindert beispielsweise, daß Maus- und Tastatur-Interrupts für eine längere Periode abgeschaltet bleiben — wobei lang genug für einen Benutzer schon beispielsweise eine Verzögerung von einer Sekunde zwischen der Bewegung der Maus und der Folgebewegung des Cursors sein kann — oder um den Ausgang des Tastatur-Kontrollers freizumachen, wenn es nötig wird, einen Befehl zum Tastatur Kontrolller zu senden.

Da die Hardware der vorliegenden Erfindung nicht dem PC/AT Standard entspricht und die Firmware weder dem konventionellen PC/AT BIOS noch dem DOS entspricht, aber das ganze System dazu ausgelegt ist, dem Benutzer zu erlauben, eine konventionelle WINDOWS®-Darstellung vor sich zu sehen und mit ihr zu interagieren, müssen einige konventionelle Hardware- Interrupts und verwandte Aufrufe durch die vorliegende Erfindung gemanaged werden. In einer bevorzugten Ausführungsform werden solche Interrupts und Aufrufe mit der Terminal-Firmware behandelt, entweder durch Emulierung oder durch Modifizierung der geeigneten Antwort

auf das eingehende Signal. Diese Antworten werden nachstehend detailliert anhand des Kernels beschrieben.

Der Kernel bringt den Prozessor in einen virtuellen 8086 Modus und stellt verschiedene Tabellen ein, wie sie zum Einfangen von I/O's an den verschiedenen Ports nötig sind. Nachdem ein Zugriff auf einen ausgewählten Port eine Ablaufunterbrechung erzeugt, wird diese Instruktion, die diese Ablaufunterbrechung erzeugt hat, zerlegt, so daß der Kernel sie korrekt verarbeiten kann, um eine PC-Kompatibilität zu emulieren, was die Übertragung verschiedener Merkmalsgruppen (sog. "Personalities") erleichtert. Weil eine ganze Subroutine für jede eingefangene I/O-Instruktion abgearbeitet werden muß, wird die I/O-Instruktion nicht so schnell ausgeführt, so daß das Abfangen von I/O's auf so wenig Ports wie möglich begrenzt wird. In einigen Beispielen werden alle oder nur ein Teil eines I/O-Ports nur soweit (mit Interrupts) unterbrochen, wie es für eine AT-Kompatibilität nötig ist. Ausgehend von der Tatsache, daß beabsichtigt ist, daß die bevorzugteste Ausführungsform der vorliegenden Erfindung eine CITRIX WinFrame-Merkmalgruppe bereitstellt, ist hervorzuheben, daß auch andere Merkmalsgruppen für die Benutzung des Terminals der vorliegenden Erfindung implementiert werden können. In einigen Beispielen kann das Terminal der vorliegenden Erfindung im Speicher vielfältige Merkmalsgruppen aufweisen, wobei das Terminal zu vielen Hosts verbunden werden kann, die unter verschiedenen Betriebssystemen laufen, und deshalb solch unterschiedliche Merkmalsgruppen erwarten. Der Benutzer oder andere können dann in der Lage sein, unter diesen Merkmalsgruppen zu wählen und zwar durch einen "Hot Key" oder eine Tastatursequenz, wobei der Kernel der vorliegenden Erfindung dann die entsprechenden Kenndaten aus dem Speicher aufruft und eine Kommunikation mit dem entsprechenden Host ermöglicht.

Die I/O-Ports, die vom Kernel für die Herstellung einer AT-Kompatibilität abgefangen werden, sind:

20h (Der Befehlsport eines AT-kompatiblen Interruptkontrollers) — Nur der "Ende des Interrupts"-Befehl wird emuliert, der am Ende jedes Interrupt-Ausführungsprogramms ausgegeben wird, damit mehr Interrupts derselben oder geringerer Priorität auftreten können. Der Kernel emuliert auch die normale Betriebsweise eines PC-kompatiblen Interrupt-Kontrollers, soweit dieser Interrupts von gleicher oder geringerer Priorität blockiert, bis der "Ende des Interrupts"-Befehl empfangen wird.

21h (Das Maskierregister des AT-kompatiblen Interrupt-Kontrollers) — Wird emuliert.

40h—43h (Zugriff zu einem AT-kompatiblen 8354 System Zeitgeber) — Die Ports werden nicht emuliert, aber abgefangen, weil einige auf AT-basierende Treiber in diese Ports schreiben. Da das Interrupt-Maskierregister nach einer bevorzugten Ausführungsform der vorliegenden Erfindung auf Port 40h gesetzt ist, würden nicht eingefangene Einschreibungen mit dem Interruptmaskierregister der vorliegenden Erfindung kollidieren.

61h (Ein Steuerport für verschieden Anwendungen in einer AT-Architektur) — Ein Lautsprecher Steuerbit wird emuliert, um es zu ermöglichen, den Lautsprecher ein- und auszuschalten.

A0h (Der Befehlsport eines zweiten Interrupt-Kontrollers eines AT-kompatiblen Rechners) — wie bei Port 20h wird der "Ende des Interrupts"-Befehl emuliert.

A1h (Das Maskierregister für den zweiten Interrupt-Kontroller (ähnlich dem Port 21h)) — Wird emuliert.

2F8h bis 2FFh (Eine Standardadresse für einen zweiten AT-kompatiblen seriellen Port) — Diese Ports werden auf 5F0h bis 5FEh versetzt, die Portadressen für den zweiten seriellen Port einer bevorzugten Ausführungsform der Erfindung. Der neue Portbereich (5F0 bis 5FE) umfaßt nur "geradzahlige" Adressen. In einer beispielhaften Ausführungsform ist ein "Byte Swapping" nicht implementiert worden und deshalb kann nur auf "geradzahlige" Bytes einer des acht-Bit SIO-Bauteils zugegriffen werden. Allerdings kann erforderlichenfalls auch ein "BYTE Swapping" implementiert werden.

378h bis 37Fh (Eine Standard Adresse für einen AT-kompatiblen parallelen Port) — Wird umgesetzt zu 6F0h bis 6FEh, welches die Portadressen für den parallelen Port einer beispielhaften Ausführungsform der Erfindung sind.

3F8h bis 3FFh (Eine Standard-Adresse für einen ersten AT-kompatiblen seriellen Port) — Werden umgesetzt zu 7F0h bis 7FEh, die die Portadressen für den ersten seriellen Port einer beispielhaften Ausführungsform der Erfindung sind.

Soweit hiermit die bevorzugten Ausführungsformen der Erfindung beschrieben sind, gibt es doch zahlreiche Abwandlungen und gleichartige Lösungen, die nicht aus dem Offenbarungsgehalt der Erfindung hinausführen. Die Erfindung ist deshalb keineswegs auf die beschriebenen Ausführungsformen beschränkt.

Patentansprüche

1. Terminal für die Anzeige von Anwendungsprogramminformation von in einer Fenster-Umgebung, gekennzeichnet durch Verarbeitungsschritte, die mit einem Personal-Computer BIOS oder Platten-Betriebssystemen (DOS = Disk Operating System) nicht voll kompatibel, jedoch für den Empfang von Fenster-Informationen angepaßt sind, welche von Programmen geliefert werden, die auf einem entfernt gelegenen Anwendungs-Server ausgeführt werden, sowie durch eine Anzeigevorrichtung für die Anzeige der Fenster-Informationen.
2. Terminal, das für eine Kommunikation mit einem aus einer Mehrzahl von Hostsystemen ausgewählten Host angepaßt ist, wobei jedes der Host-Systeme ein unterschiedliches Kommunikationsprotokoll benutzt, umfassend:
 - eine erste Merkmalsgruppe zur Bereitstellung von Antworten auf einen ersten und einen zweiten Satz von Anweisungen, die von einem ersten Hostsystem gemäß dem Kommunikationsprotokoll geliefert werden, welches zu dem ersten Hostsystem gehört, zumindest eine zweite Merkmalsgruppe zur Bereitstellung von Antworten auf einen dritten und einen vierten Satz von Anweisungen, die von einem zweiten Hostsystem gemäß dem Kommunikationsprotokoll geliefert werden, welches zu dem zweiten Hostsystem gehört, wobei zumindest eines der Hostsysteme Fenster-Informationen bereitstellt, wobei weiterhin Prozessorvorrichtung-

gen vorhanden sind, die auf die erste und zumindest auf die zweite Merkmalsgruppe zum Verarbeiten eines des ersten und dritten Satzes von Anweisungen und zum Emulieren eines des zweiten und vierten Satzes von Anweisungen ansprechen, zum Bereitstellen von Antworten an das zugehörige Hostsystem, die mit dem Kommunikationsprotokoll, das zu jenem Hostsystem gehört, kompatibel sind, und
 Schaltvorrichtungen zur Identifizierung des Hostsystems, mit welchem das Terminal gegenwärtig verbun- 5
 den ist, und zur Auswahl des geeigneten der ersten und zumindest der zweiten Merkmalsgruppe für Kommunikationen mit dem identifizierten Hostsystem.

3. Verfahren zum Aktualisieren von Betriebscharakteristiken eines Terminals, folgende Schritte umfassend:
 Verfügbarmachen eines Flashspeichers zur Speicherung von Betriebseigenschaften,
 Bereitstellen einer Kommunikationsverbindung zu einem Host für den Empfang von aktualisierten Be- 10
 triebscharakteristiken,
 Herstellen eines ersten Betriebszustandes, während dem zumindest in einen vorbestimmten Teil des Flashspeichers eingeschrieben werden kann, während die Kommunikation mit dem Host aufrechterhalten bleibt,
 Herunterladen der aktualisierten Betriebscharakteristiken vom Host auf das Terminal, während sich das 15
 Terminal im ersten Betriebszustand befindet,
 Einschreiben der aktualisierten Betriebscharakteristiken in den Flashspeicher, während sich das Terminal im ersten Betriebszustand befindet, und
 Herstellen eines zweiten Betriebszustandes, während dem Daten nicht in den vorbestimmten Teil des Flashspeichers geschrieben werden. 20

4. Terminal zur Anzeige — in der Microsoft WINDOWS®-Betriebsumgebung — von Anwendungspro-
 gramminformationen, welche von einem Hostsystem, auf welchem ein Microsoft WINDOWS®-Betriebssy-
 stem läuft, bereitgestellt werden, mit
 einer Schnittstellenvorrichtung zum Empfang von Anzeigeinformation vom Anwendungs-Server für ein 25
 Anwendungsprogramm, das auf dem Anwendungs-Server abläuft,
 Verarbeitungsvorrichtungen, die nicht voll mit den Personalcomputer-BIOS oder Platten-Betriebssystemen kompatibel sind und zur örtlichen Durchführung des Anwendungsprogramms unfähig sind, und auf die Schnittstellenvorrichtungen ansprechen, entweder zum Emulieren oder zum Verarbeiten von Anweisun-
 gen, die von Programmen zugeführt werden, die auf einem entfernten Anwendungs-Server ausgeführt 30
 werden, um dem Hostsystem Antworten bereitzustellen, die mit denen kompatibel sind, die vom Hostsystem erwartet werden, und
 Anzeigevorrichtungen, die auf die Verarbeitungsvorrichtungen ansprechen, um die Fenster-Informationen anzuzeigen.

5. Verfahren zum Konfigurieren eines Terminals mit einer Anzeige für die Kommunikation mit einem Hostsystem, mit folgenden Verfahrensschritten: 35
 Herstellen zumindest eines Punktraster- (bit-mapped) Bereichs innerhalb eines Bereichs der Anzeige,
 Herstellen zumindest einer Punktraster- (bit-mapped) Gruppe innerhalb des Bereichs, und
 Herstellen zumindest einer Punktraster- (bit-mapped) Auswahl innerhalb der Gruppe, wobei mit jeder Auswahl eine oder mehrere Wahlmöglichkeiten einhergehen. 40

Hierzu 9 Seite(n) Zeichnungen

40

45

50

55

60

65

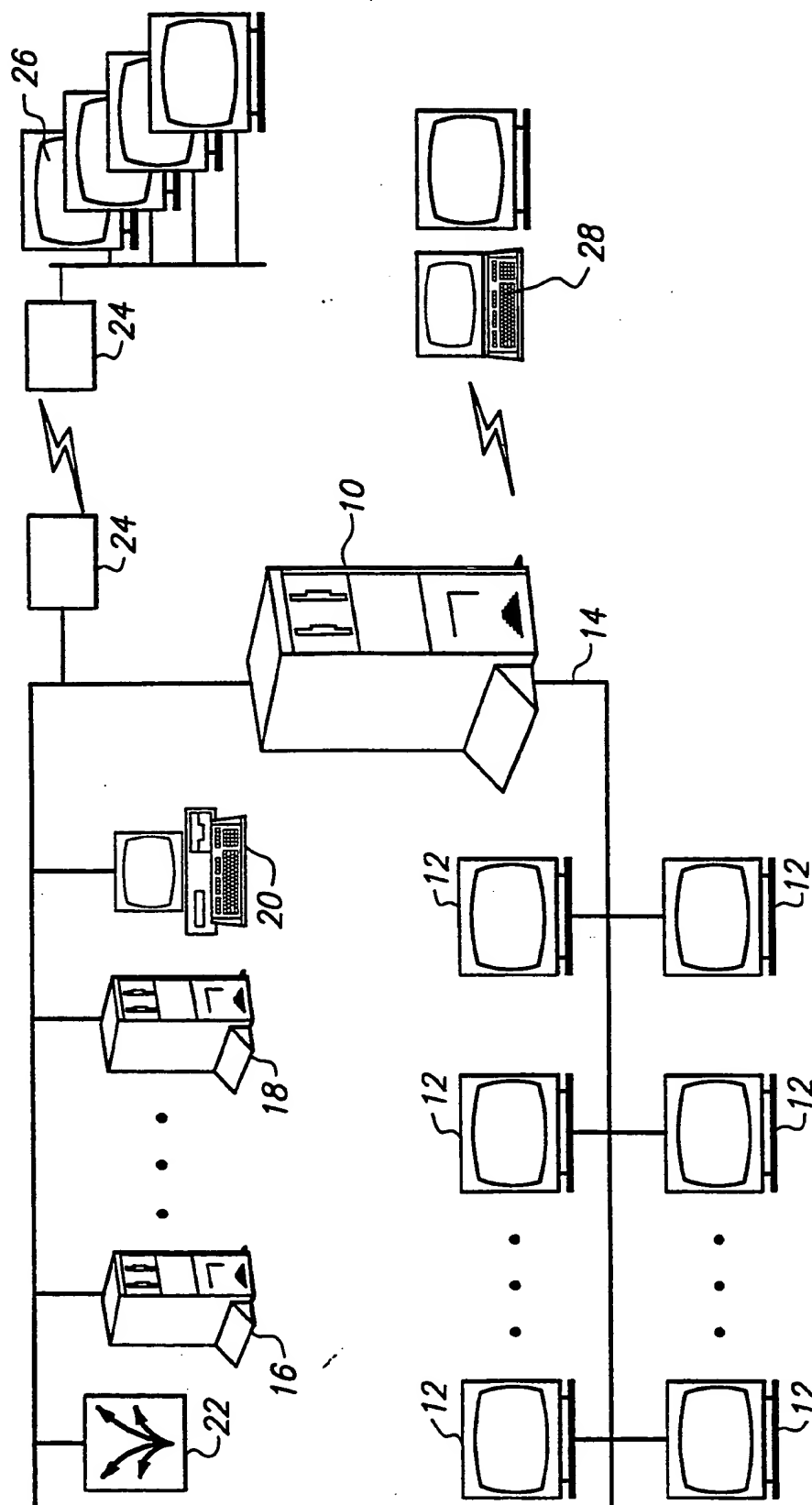


FIG. 1

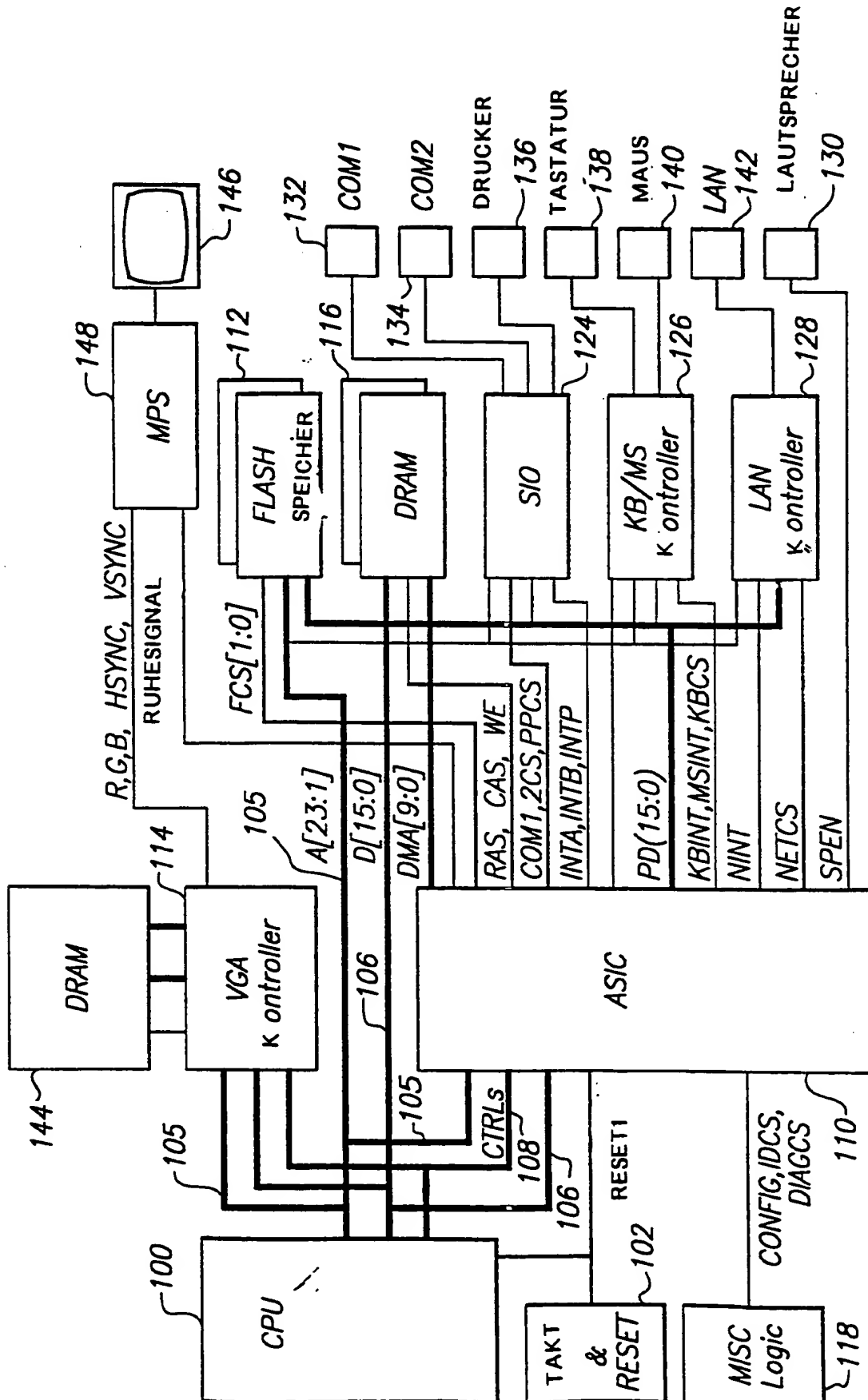


FIG. 2

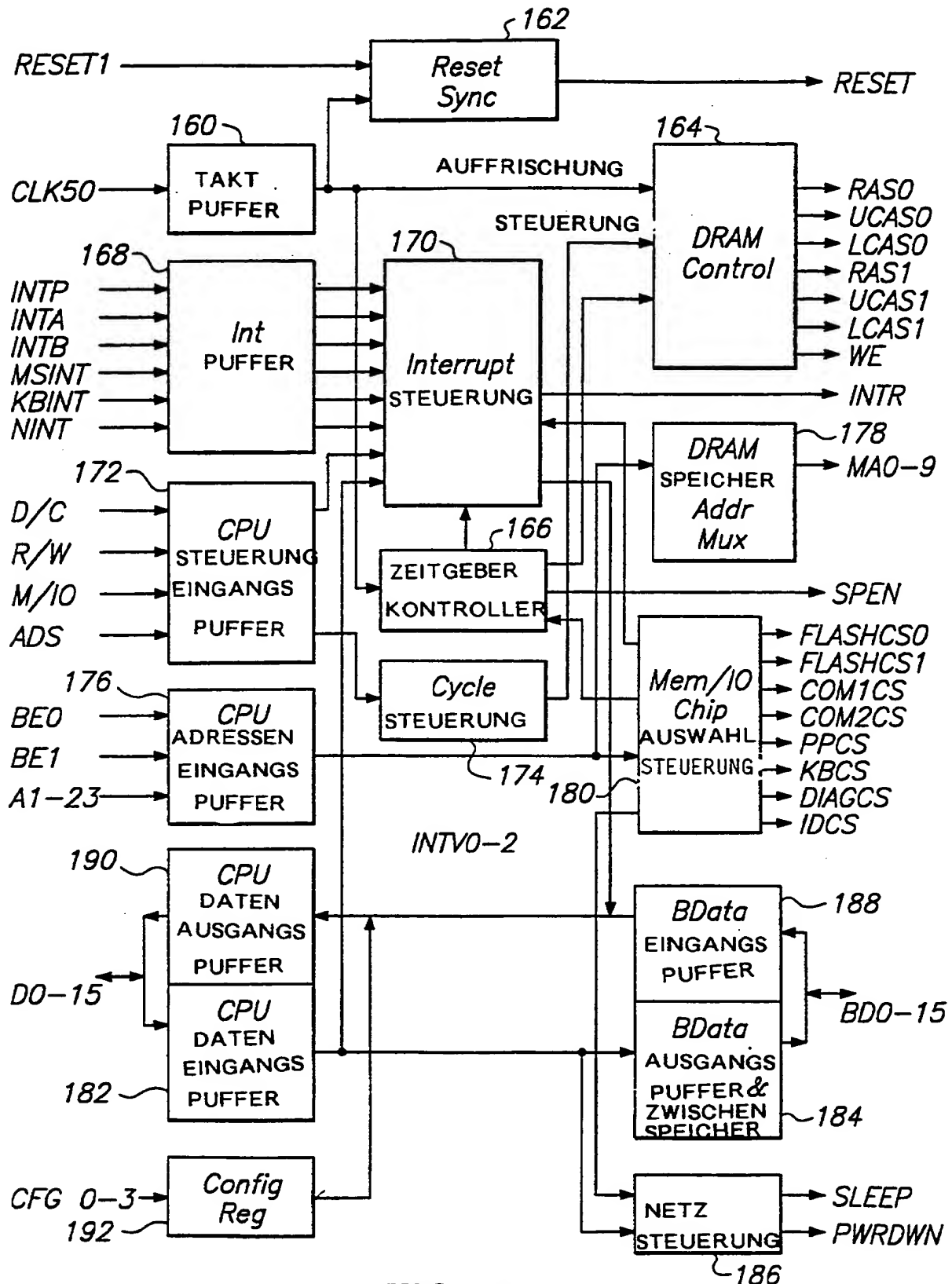


FIG. 3

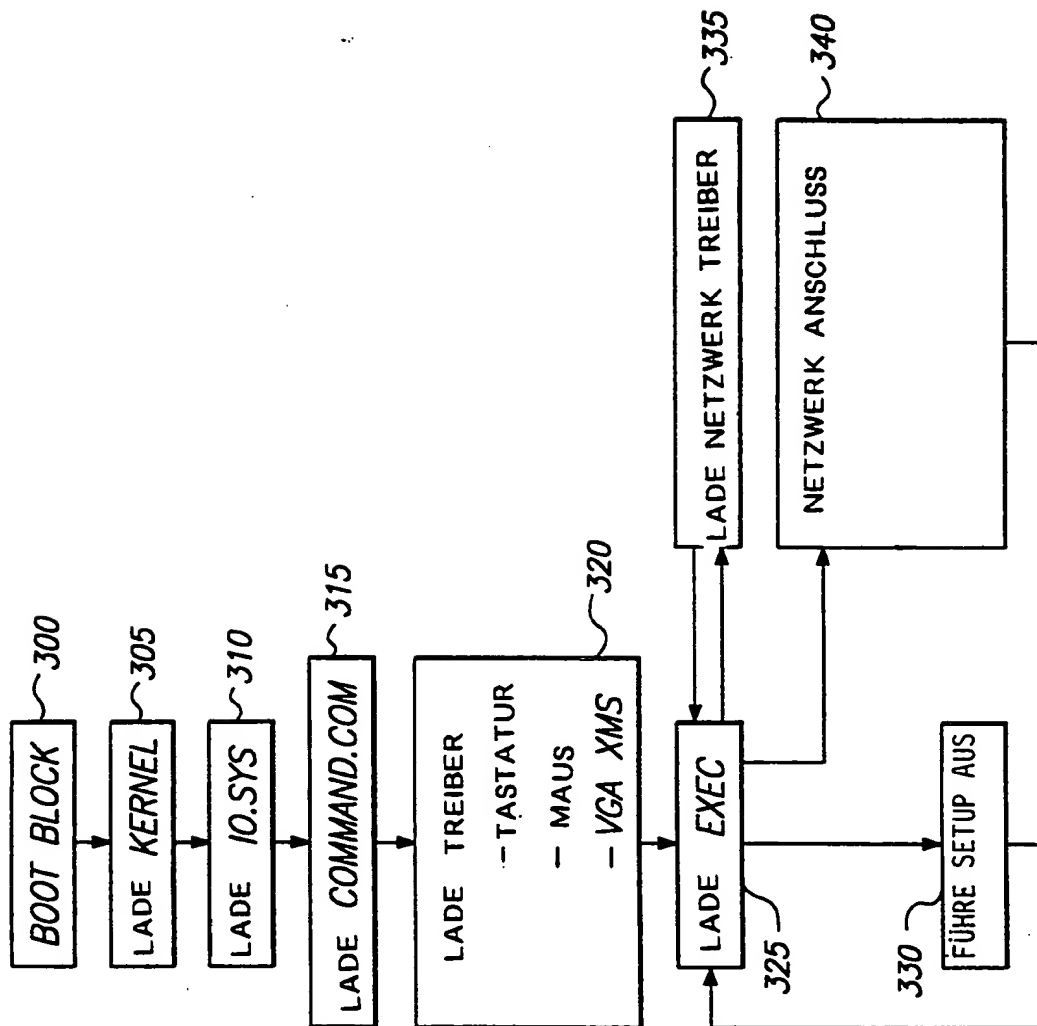
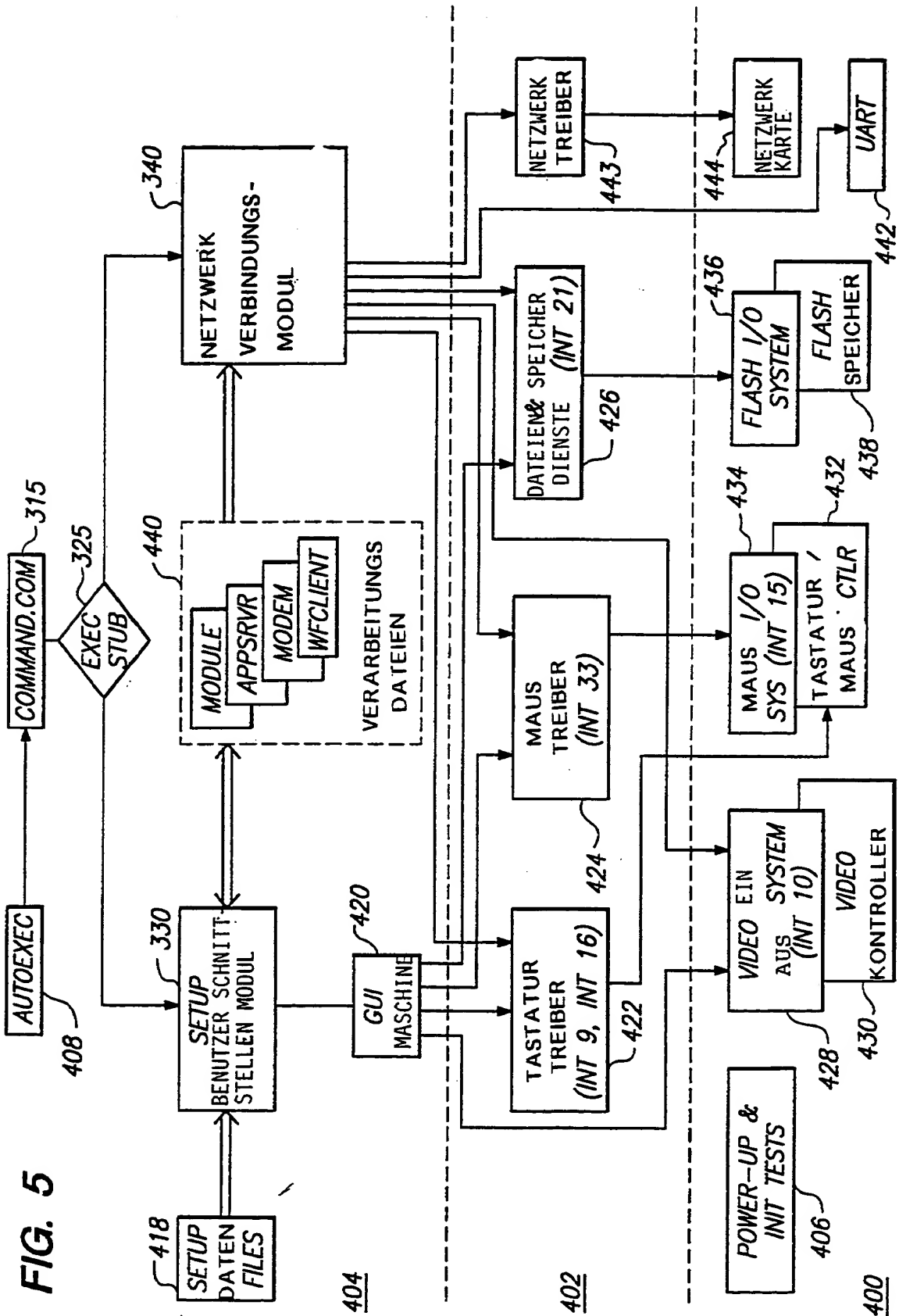


FIG. 4



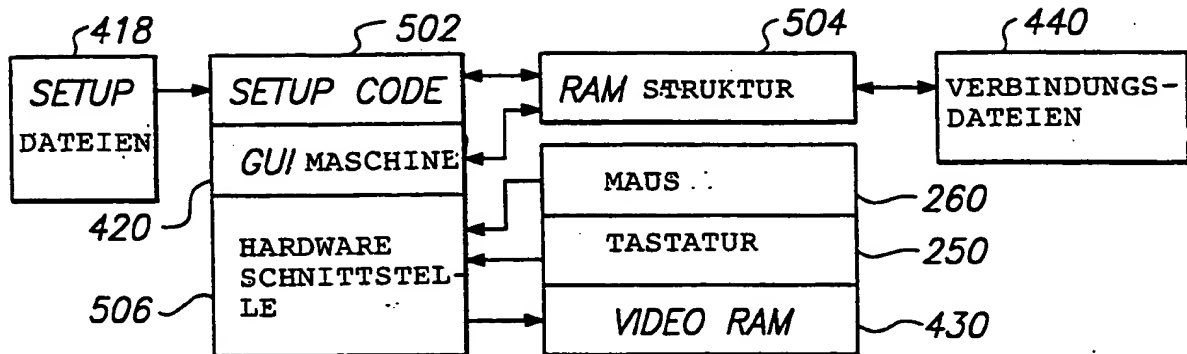


FIG. 6

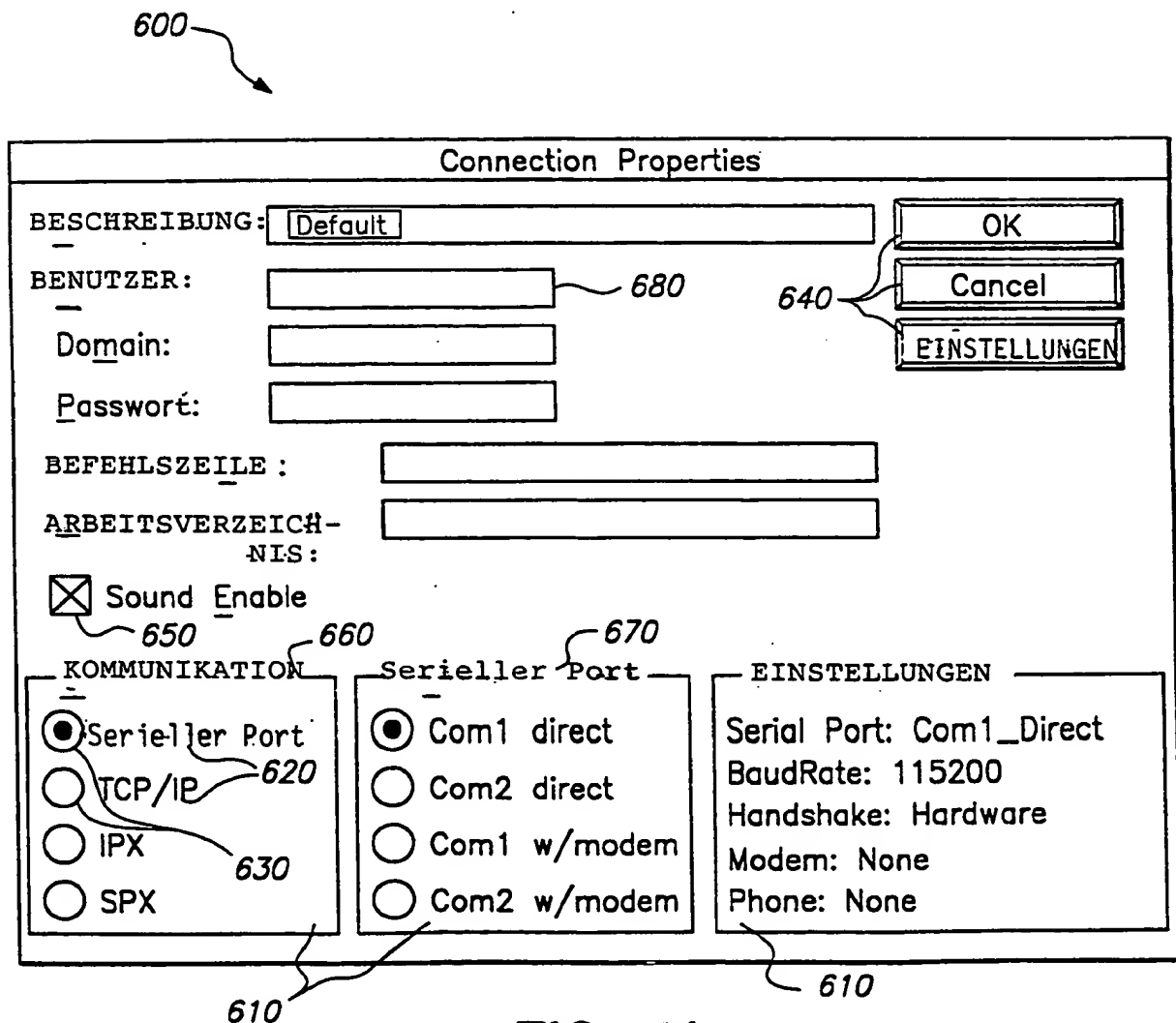


FIG. 7A

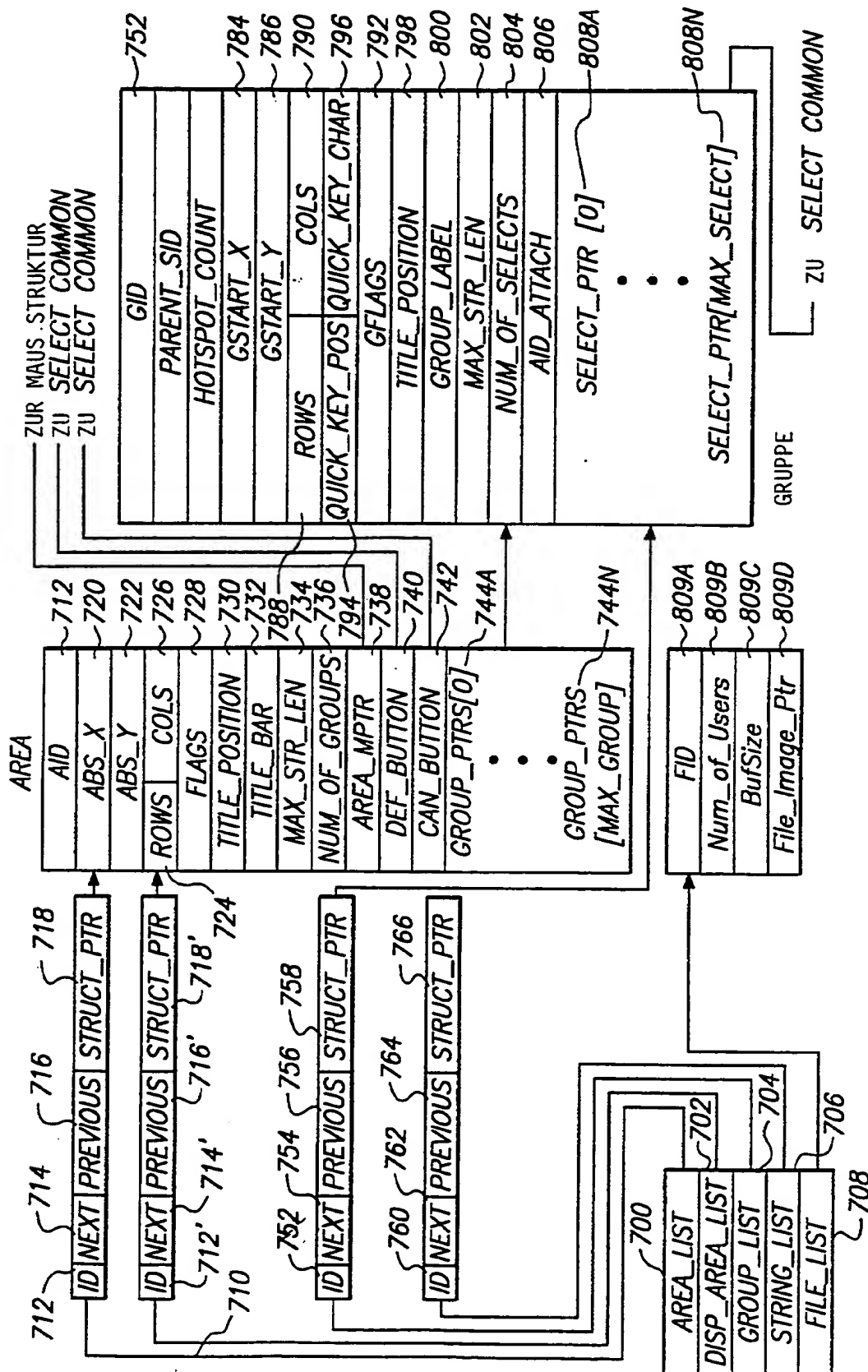
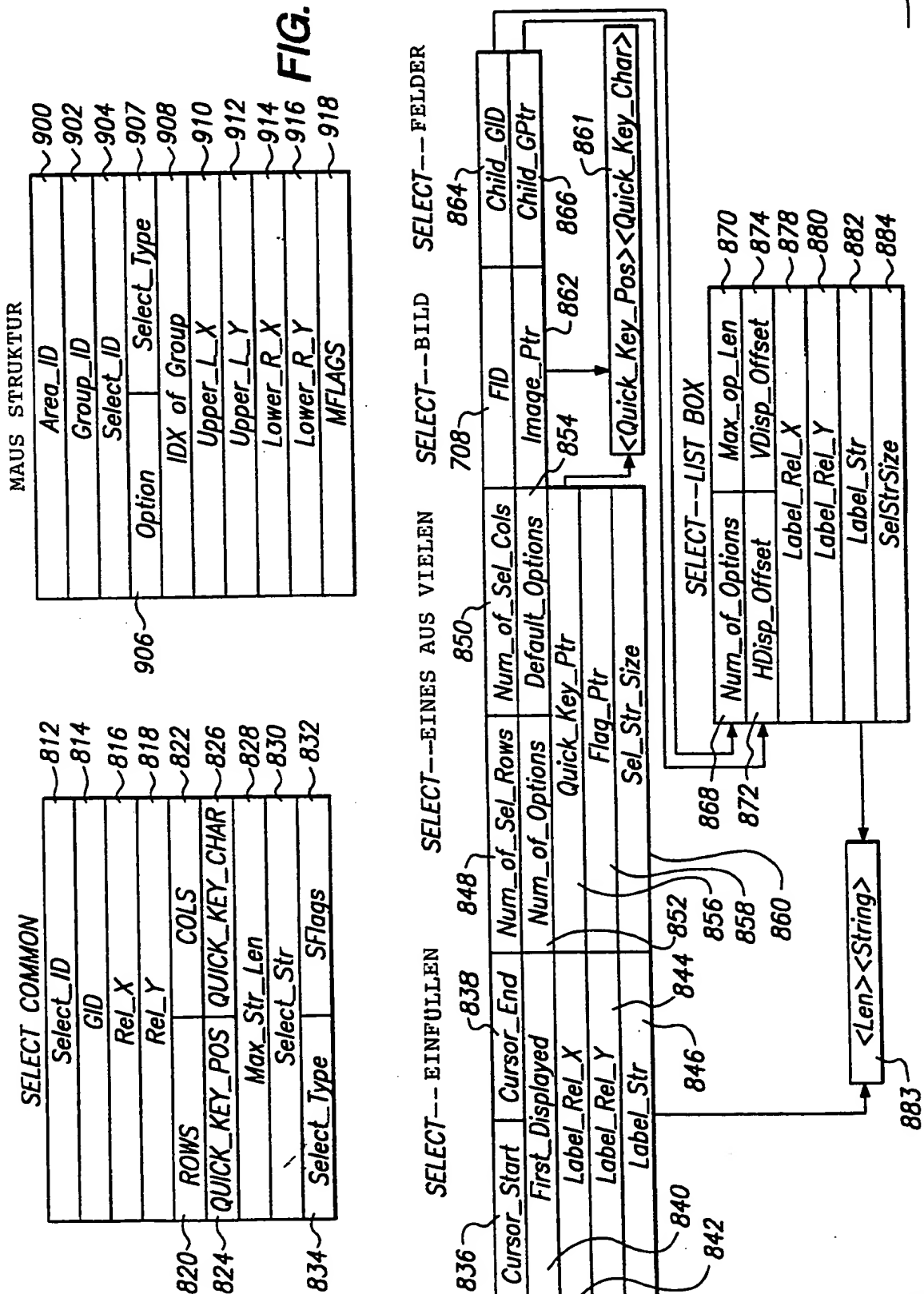


FIG. 7B1

FIG. 7B2



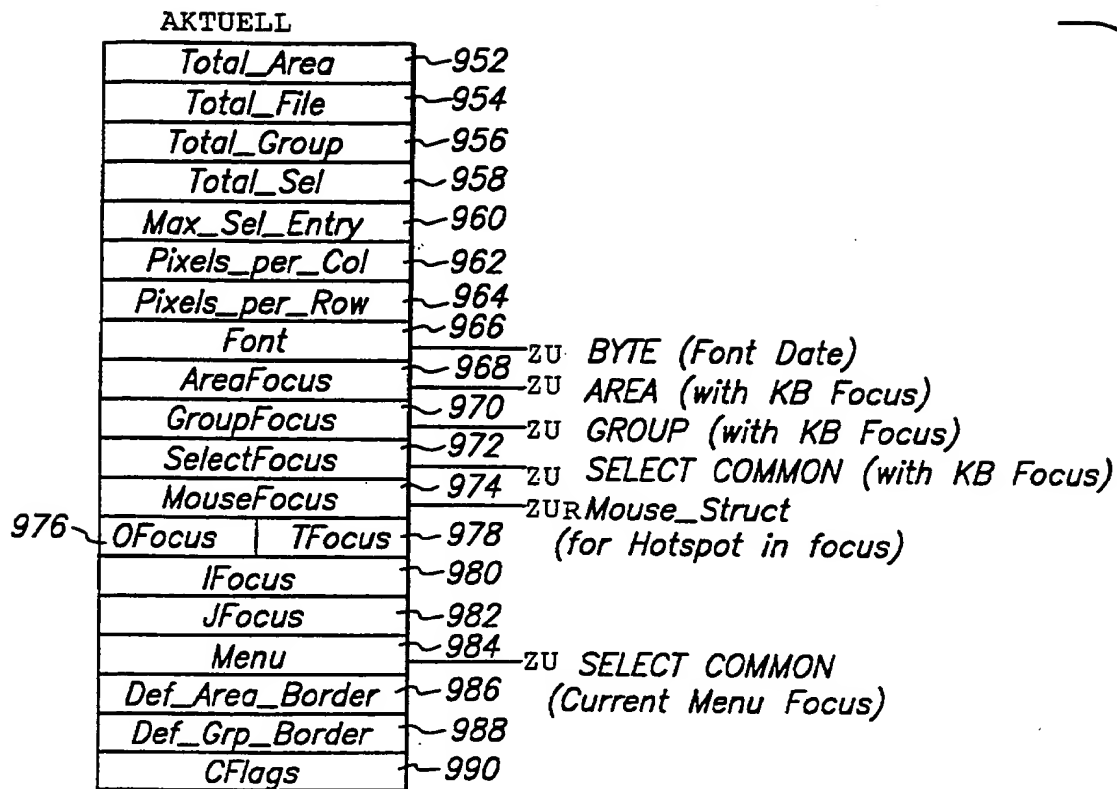


FIG. 7B3

